NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A FORMAL ANALYSIS OF THE MLS LAN: TCB-TO-TCBE, SESSION STATUS, AND TCBE-TO-SESSION SERVER PROTOCOLS

by

Daniel Shawn Craven

September 2004

Thesis Advisor:	George W. Dinolt
Second Reader:	Sylvan S. Pinsky

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: **A Formal Analysis of the MLS LAN: TCB-to-TCBE, Session Status, and TCBE-to-Session Server Protocols**			**5. FUNDING NUMBERS**
6. AUTHOR(S) Daniel S. Craven			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

This thesis presents a formal analysis process and the results of applying that process to the MLS LAN: TCB-to-TCBE, Session Status, and TCBE-to-Session Server Protocols. The formal analysis process consists of several distinct stages: the creation of a detailed informal protocol description, analyzing that description to reveal assumptions and areas of interest not directly addressed in the protocol description, the transformation of that description and the related assumptions into a formal Strand Space representation, analyzing that representation to reveal assumptions and areas of interest, and concluding with an application of John Millen's automated Constraint Checker analysis tool to the Strand Space representations under an extremely limited set of conditions to prove certain protocol secrecy properties.

14. SUBJECT TERMS Protocol Analysis, Constraint Checker, Strand Spaces			**15. NUMBER OF PAGES** 155
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

A FORMAL ANALYSIS OF THE MLS LAN: TCB-TO-TCBE, SESSION STATUS, AND TCBE-TO-SESSION SERVER PROTOCOLS

Daniel S. Craven
Civilian, Federal Deposit Insurance Corporation
B.A., University of California Santa Barbara, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2004

Author: Daniel Shawn Craven

Approved by: Dr. George W. Dinolt
Thesis Advisor

Sylvan S. Pinsky
National Security Agency
Co-Advisor

Dr. Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis presents a formal analysis process and the results of applying that process to the MLS LAN: TCB-to-TCBE, Session Status, and TCBE-to-Session Server Protocols. The formal analysis process consists of several distinct stages: the creation of a detailed informal protocol description, analyzing that description to reveal assumptions and areas of interest not directly addressed in the protocol description, the transformation of that description and the related assumptions into a formal Strand Space representation, analyzing that representation to reveal assumptions and areas of interest, and concluding with an application of John Millen's automated Constraint Checker analysis tool to the Strand Space representations under an extremely limited set of conditions to prove certain protocol secrecy properties.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No.DUE-0114018.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

I would like to thank my thesis advisor Dr. George W Dinolt. His insight and guidance were invaluable. I am indebted to him not only for his generosity with his time but also for his ability to allow me to struggle with difficulties yet keep me from "losing my way".

I would also like to thank the Center for INFOSEC Studies and Research, J D Wilson and the members of the MLS LAN development team, Dr. Cynthia Irvine and the Monterey Security Architecture project (MYSEA) and Dr. Sylvan S. Pinsky, Nation Security Agency. Each of the aforementioned contributed to the completion of this thesis. I am grateful to each and everyone.

Additionally, I would like to thank my family and friends who have been supportive during this entire endeavor.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PURPOSE OF STUDY

"The MLS LAN Project is an effort to provide government and commercial organizations with a cost effective, multilevel networking solution by leveraging existing high assurance technology".[1] Because of the requirements of this communications framework, the protocols that are associated with this framework must not only be secure, but must be demonstrably so. To this end, this thesis will attempt to answer some important questions about the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols. One of the most important questions is: How sound are the protocols with respect to the security polices that they are expected to enforce and work within?

To answer this requires a methodology that will increase the confidence of both internal developers and outside inspectors of the MLS LAN Project that the protocols implement the security policies of the MLS LAN. This thesis will follow a series of steps that will do just that. By developing a semiformal representation of the MLS LAN security policies, the MLS LAN TCB-TCBE Connection, Session Status, and TCBE-to-Session Server Connection protocols, this thesis will build on the foundation of previous work and more concisely define the specifications. These semiformal representations will in turn, support the construction of a more formal specification of the protocols that can take advantage of the protocol analysis tools and techniques that are currently available.

Using the formal specifications and the formal protocol analysis techniques, we will analyze the protocols to provide higher assurance that they meet the desired security properties and to identify possible weaknesses in the protocols

B. ORGANIZATION OF PAPER

This paper is organized into seven parts. The first section is a simple introduction to the goal of this paper. The second section, entitled background, introduces general background information starting with a definition of the term protocol, an explanation of why protocols are important, and why formal protocol analysis is a worthwhile endeavor.

The same background section continues with an extremely brief survey of the field of cryptographic protocol analysis.* Followed by an introduction to the Multilevel Secure Local Area Network (MLS LAN) project.[1] The background section concludes with a description of the items that will be analyzed and how that material is presented. The third section describes the methodology that drives the work presented in this paper. The fourth section describes the protocol specifications for each of the analyzed protocols. This information is based on the comprehensive information unit mapping, which is described and presented in appendix A. The presentation of the protocol specification information includes the requirements placed on the protocol, the entities authorized to participate in the protocol, the authorized packets, state transitions, and any relevant additional information. The fifth section introduces the formal protocol analysis technique used; Strand Spaces. This section continues by presenting the formal properties of the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols expressed in Strand Space notation. This information is based on the work presented in Appendix B. The sixth section presents the results of the three stages of analysis, which correspond to the work in Appendix A, B, and C, respectively. The final section presents a conclusion of the work presented.

* Information in "Important Developments in Formal Protocol Analysis" is heavily based on very detailed papers written by Meadows and other experts in the field. These papers are noted in the appropriate locations and the reader is urged to consult them.

II. BACKGROUND

A. PROTOCOL DEFINITION

There are various definitions of the word protocol. Webster's Revised Unabridged Dictionary gives one definition of protocol as "A preliminary document upon the basis of which negotiations are carried on"[2]. The American Heritage Dictionary's definition includes "A code of correct conduct"[3] as one of the possibilities. Finally, a computer specific dictionary gives as part of its definition "A set of formal rules describing how to transmit data, especially across a network."[4] All of these definitions are correct. While this paper will focus on the more computer centric definition of protocol, it is more important for now to simply realize that protocol is just another word for "communication framework". A protocol is a way to communicate, using a set of rules that the participants know.

B. IMPORTANCE OF PROTOCOLS

Protocols are important because they are everywhere. A good example of a protocol that people use everyday is a normal phone conversation. The "phone call protocol" follows a typical protocol pattern. The participants make a connection. They authenticate each other. They exchange information and then they terminate the connection. This is just one of the many protocols people use everyday. People use protocols for one simple reason:

- Protocols make communication more effective.

Protocols make communication more effective because they allow participants to make assumptions about information. Most people use these assumptions subconsciously because when protocols are used as intended by honest participants they are almost invisible to the participants. The participants use the assumptions that are associated with the protocol and simply focus on the information. However, protocol effectiveness has a price. The assumptions must be valid. The only way to establish the validity of the assumptions is through some type of analysis.

The following summary, of an informal analysis of one part of the "phone call protocol" mentioned earlier, illustrates how even informal protocol analysis can illuminate aspects of a protocol that might be otherwise be unrealized.

When a phone conversation ends the participants don't just hang-up. Surprisingly, the participants don't just say goodbye and hang-up. An informal analysis showed that there is a "pre-goodbye" that is sent and acknowledged. Normally the "hang-up" initiator sends a pre-goodbye indicator; for example "well, I should get going" or "It was nice talking to you". The actual phone conversation termination is as follows. A "pre-goodbye" is sent and acknowledged. An actual goodbye is sent and acknowledged. Then the participants terminate the connection.

It is interesting to note that the absence of the "pre-goodbye" often causes confusion in one of the participants. This is a trivial example done in an informal manner and has no scientific value for this paper. However, it does illustrate how even informal protocol analysis can discover aspects of protocols that may not have been understood prior to the analysis.

An interesting side note is that many successful situational-comedies are based on protocol analysis. They normally develop as follows. Someone receives some information and makes an erroneous assumption that leads to a humorous situation. The resolution is when someone points out the erroneous assumption. A perfect example of simple informal protocol mis-analysis!

C. IMPORTANCE OF FORMAL PROTOCOL ANALYSIS

Formal protocol analysis is difficult. Needham and Schroeder[5] are credited with first stating that fact and inadvertently proving it. Cathy Meadows also believes that "security flaws in a protocol can be subtle and hard to find".[6]

A perfectly natural question is: why is protocol analysis important? Even if there were only honest participants using protocols, protocol analysis would still be important because without analysis there is no way to know for sure what the assumptions used in the protocol actually are. Assumptions cover items such as who the participants are, how

certain pieces of information should be treated, and the properties that are enforced by the protocol. Additionally protocol analysis helps illuminate the assumptions used by the protocol itself and those used by the participants as either valid or erroneous. Protocol analysis does not have to be formal to be beneficial. In fact, people are continually informally analyzing protocols. I don't propose that we formally analysis the "phone call protocol". What I do propose is that protocols that are used for computer communications need to be analyzed. Computers don't have the ability to correctly evaluate information they receive when the assumptions based on the protocol that delivered the information are erroneous. The need for formal protocol analysis grows as the level of trust placed on the system increases. For example, in a multilevel secure system, when a protocol delivers two separate pieces of information - the user name Foo and the label Top Secret - the assumption is that the user Foo is able to read Top Secret information. That is an important association.

D. IMPORTANT DEVELOPMENTS IN FORMAL PROTOCOL ANALYSIS

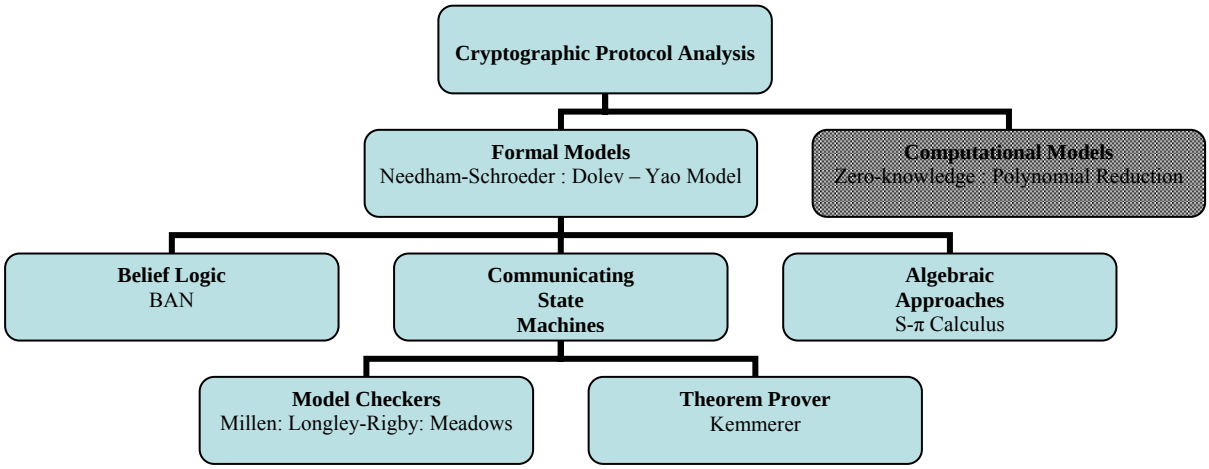


Figure 1. Important Developments in Formal Protocol Analysis (Adapted From Ref 6,7,8)

1. Cryptographic Protocol Analysis

There are two schools in cryptographic protocol analysis. The first, called computational models focuses on the security of the cryptographic algorithms themselves. It uses techniques such as zero-knowledge and polynomial reduction to analyze the algorithm's complexity-theoretic properties.[6] The second school and this paper are concerned with the logical interaction of the participants of the protocol, independent of the cryptography used in the protocol.

2. Formal Models

a. Needham and Schroeder

The paper "Using Encryption for Authentication in Large Networks of Computers" published in 1978 by Needham and Schroeder[5] is considered by many to be the start of any discussion of protocol analysis. This paper discussed three protocols. The first of these was a protocol with the goal of establishing interactive communication between two authenticated principals. The second protocol was an authenticated one-way communication. The final protocol dealt with signed communication. The impact of this paper on protocol analysis as a whole was not in the protocols themselves. The real impact was the notion that protocols "are often subject to non-intuitive attacks which are not easily apparent even to a careful inspection"[9]. The Needham and Schroeder paper inadvertently gave two examples of protocols that received extensive hand analysis by experts and were generally considered sound yet still contain weaknesses.[10,11] Additionally, the Needham and Schroeder paper is often sited as stating that formal methods could be used to assure correctness. While this may have been the intent the paper actually states that protocols "are prone to extremely subtle errors that are unlikely to be detected in normal operation. The need for techniques to verify the correctness of such protocols is great".[5] How right they were. Ironically, formal methods were later used to show that both the authenticated connection protocol and the authenticated mail protocol had weaknesses.[10,11] The fact the paper directly addresses, that creating and analyzing protocols is difficult, is why it is considered the start of most discussions about the subtly and complexity of protocol analysis.

b. Dolev and Yao

The next important step was the development of a formalization of the intruder model by Dolev and Yao.[12] This was an important step because it was the first formal model of an environment that had three distinct characteristics. First of all, multiple executions of the protocol could be running concurrently. Secondly, the cryptographic algorithms were treated as "black boxes" which obeyed a limited set of algebraic properties. Lastly, and most importantly, was the model of an intruder that had the ability to read, alter, create, and destroy traffic as well as control some of the legitimate members of the system.[12] This formalization of the intruder, or some variation of it, is used in most of the protocol analysis work done today.[9] The model assumes several things about the abilities of the penetrator:

The penetrator controls the network to the point that all traffic can be considered sent to the penetrator and received from the penetrator. The penetrator can create messages as a legitimate user of the protocol as well as prevent or alter the messages of legitimate users. The penetrator, equipped with the appropriate key, has the ability to encrypt and decrypt messages. The penetrator can make random choices and create new keys. The penetrator "can not guess a random number which is chosen from a sufficiently large space".[13] The penetrator cannot guess a cryptographic key that the penetrator does not have access to through information sent across the network.

The abilities of the penetrator in this model contribute to the difficulty of protocol analysis. "Most of the work that has been done on applying formal methods to cryptographic protocols has relied upon the Dolev-Yao model".[14] While the descriptions of the penetrator's abilities are simple, the state space of possibilities quickly explodes. This paper will also use the Dolev-Yao intruder model. Formal models based on the Dolev-Yao intruder model fall into three general areas: Belief logic, Communicating state machines, and Algebraic Approaches.

3. Belief Logic

A major area of research in the application of formal methods to protocol analysis is in the area of belief logic.[6,15] This is very similar to the application of modal logics that have been applied in distributed systems.[6] In contrast to communicating state

machines, belief logics concern themselves with statements about belief. These statements about belief are based on an initial set of beliefs. As messages are received, beliefs are added to the initial set. The initial set of beliefs is also expanded using induction. At the end of the protocol, if the set of beliefs is "adequate" then the protocol is assumed to be correct.

a. Ban Logic

The goal of BAN logic is to define a logic of authentication to express:

- What principals should be entitled to believe
- Express those beliefs precisely
- Capture the reasoning that leads to those beliefs

"The intended use of BAN is to analyze authentication protocols by deriving the beliefs that honest principals correctly executing a protocol can come to as a result of the protocol execution."[16] The goals of authentication are stated as follows: "After authentication, two principals (people, computers, services) should be entitled to believe that they are communicating with each other and not with intruders."[17] BAN logic attempts to address the problems of protocol analysis that relate directly to authentication protocols. "Although authentication protocols typically have few messages, the composition of each message can be subtle, and the interactions between the messages can be complex."[17] BAN logic uses a logical syntax that has an intuitive structure. This syntax contains several objects such as principals, encryption keys and statements. These are used to construct statements such as the following:

- "P believes X"
- "P sees X"
- "P said X"

While BAN logic has been used to find previously unknown weaknesses in several protocols[17] there are several areas that it doesn't address. BAN logic doesn't have a formal semantic, nor a formal adversary, doesn't address dishonest participants, different levels of trust and assumes perfect cryptography. BAN Logic can be used for authentication proofs, but it doesn't address confidentiality. While BAN Logic has many strengths, is also has some weaknesses.[18] Several other belief logics have been developed to address some of these areas. Many of them are based on BAN logic

constructed by Burrows, Abadi, and Needham.[17] These include: GNY[19], BGYN[20], SvO[21], Kailar's Logic of accountability[22], and Wedel and Kessler's Logic[23].

. Communicating State Machines

Communicating State Machines are often used in the analysis of cryptographic protocols that incorporate the Dolev-Yao model of the penetrator. Each protocol participant is modeled as a state machine which transitions state based on communications sent to and received from other participants of the protocol.

5. Model Checkers

Model checking techniques attempt to create a finite model of protocol that reflects the security properties the protocol attempts to provide. Then the model is "checked" to verify that the property is satisfied[24]. The one of the main challenges of model checking is containing state space explosion. Two of the most well known model checking tools are described below.

a. Millen

Jonathan Millen's Interrogator model is a security analysis tool that is based on "communicating machine transformation model with message modification threats." [25] His automated tool, written in prolog, uses an exhaustive search of the protocol participant's state space to attempt to locate protocol security flaws.[6] Other similar tools have incorporated human interaction in an attempt to enhance the tools abilities. On such example is the tool developed by Longley-Rigby.[26]

b. Meadows

Cathy Meadows is one of the most important individuals in protocol analysis today. Working at the Naval Research Laboratories (NRL) she and her staff have made many contributions to the field. One of the most important of these is the NRL Protocol Analyzer[27]. The NRL Protocol Analyzer is a formal methods tool that models specified protocols as communicating state machines. One of the state machines is a representation of a version of the Dolev-Yao model. The tool is used to check insecure states. Some of the strengths of the NRL protocol analyzer are: Effectively represents the Dolev-Yao intruder, by making no assumptions about the number of: protocol executions, principals performing the different executions, interleaved executions, or times cryptographic functions are applied.[9]

6. Theorem Prover

In theorem proving the protocol itself and the desired properties of the systems are expressed in a formal logic. Then using a set of axioms and inference rules the properties of the system can be proven or refuted.[24]

a. Kemmerer

Richard Kemmerer's approach is to use a "conventional formal specification language"[6], specifically Ina Jo[28,29] In doing so he not only is able to reap the benefits of the model checker but he is able to prove properties about the system using standard theorem proving techniques.

E. MULTILEVEL SECURE LOCAL AREA NETWORK PROJECT

Full coverage of the Multilevel Secure Local Area Network Project (MLS LAN) and its goals are presented in J D Wilson's master thesis entitled: A Trusted Connection Framework for Multilevel Secure Local Area Networks.[1] While that effort will not be repeated here, the following section will highlight the major issues and goals of the MLS LAN project as presented in Mr. Wilson's thesis.

Most people are aware of the military's use of the Unclassified, Confidential, Secret, and Top Secret multilevel system of classification. However, many people don't realize the commercial world's own need for a multilevel system. If the Coca Cola Company only had one level of classification for information there couldn't be a "Secret Recipe". The commercial world's labels may be the same as the military's labels or they may be different, such as "non-proprietary" and "proprietary" but the underlying need is the same. That need is for a system that "enables an organization to maintain a single network that is sufficient to verifiably restrict access to only that data for which the user is both cleared and has the requirement to see, even though the network contains data at multiple sensitivity levels".[1] In the days of paper systems this was relatively straightforward. Someone was responsible for the documents and the appropriate security measures were used to store and distribute the information. They knew who

could access what and checked the items in and out. Since the transition from the paper based system to the electronic system there have been several attempts to design systems that gave the same assurances.

While there are other solutions, the "Dedicated", "System High" and "Compartmented" systems, they all fail because they are too expensive.[1] Expensive is defined in this paper as the total combination of time, cost, and difficulty of redundant hardware, system administration, infrastructure management, specialized hardware, specialized software, or inappropriate security level granularity. The MLS LAN is a proposed solution to this problem.

"The MLS LAN Project is an effort to provide government and commercial organizations with a cost effective, multilevel networking solution." [1] The MLS LAN has grown out of research started in 1997 at the Center for Security and Information Security (INFOSEC) Studies and Research (CISR) at the Naval Postgraduate School (NPS) in Monterey California. It is a project that is attempting to build a multilevel secure network that leverages the use of existing high assurance technology and commercial off the shelf products (COTS) to help minimize the expense of the system; which has been the main inhibiting factor in previously developed multilevel secure systems. The project uses a small number of verified high assurance stand alone systems as the basis for the multilevel high assurance network that provides services and data to inexpensive "diskless" workstations. The MLS LAN provides several guarantees. These are that the MLS LAN "maintain absolute control over the mechanism that provides data to the users" [1] and that the MLS LAN be able to "verifiably ensure the identity and coinciding security factors associated with each user accessing the network."[1] Additionally, The MLS LAN project allows "for independent evaluation under an accepted standard criterion".[1]

The MLS LAN framework strives to provide protected communications among each of the components of the MLS LAN and to allow users to negotiate session level privileges within a multilevel secure system.[1] That framework consists of the following four protocols:

- Protected Communications Channel (PCC)
- Trusted Computing Base to Trusted Computing Base Extension Connection (TCB-to-TCBE)
- Session Status
- Trusted Computing Base Extension to Session Server Connection (TCBE-to-Session Server)

F. WHAT WILL BE ANALYZED?

The main goal of this paper is to analyze the three protocols: the TCB-to-TCBE Connection, Session Status, and TCBE-to-Session Server protocols. All three of the protocols rely on the conduit established by the PCC. Presently, the PCC is a stock implementation of IPSec and therefore this paper will not attempt to formally analyze IPSec. This paper will focus on the three protocols that depend on the PCC. These protocols will be analyzed in order to increase the confidence in the completeness and necessity properties of the protocols themselves and to explicitly express the assumptions the protocols impose on the PCC channel.

G. DOCUMENT STRUCTURE

This chapter has presented a general introduction to both the need for protocol analysis and the important developments in the field. The following chapter presents an overview of the methodology used in this application of protocol analysis. Chapter III presents the process used in mapping the original presentation of the protocols to a semi-formal representation. Chapter three also presents relevant findings that are discovered at this stage in the process. The next chapter takes the semiformal representations from chapter three and presents a mapping between the different abstraction levels. Chapter V gives a general description of formal Strand Space models and then presents the Strand Space representations of the three analyzed protocols, along with issues that arose at this stage of the analysis. The final chapter presents a summary of the findings from each stage of the analysis, as well as conclusions and areas of future work.

III. METHODOLOGY

The main goal of this paper is to analyze the three protocols: the TCB-TCBE Connection, Session Status, and TCBE-to-Session Server protocols. In order to address the primary goal this paper presents a methodology of how apply protocol analysis. A legitimate question is why is this process necessary. The process is necessary it allows one to discover properties about the protocols that might not otherwise be discovered.

In addition the development and application of the process allows one to:

- Highlight aspects of the protocols that could benefit from an increase in documented specification details.
- Provide a simple process that could be used repeatedly during the development process to illustrate areas of interest.
- Present a process that enhances the ability to prove the properties of the system.
- Present a process that can prove that the system, given the assumptions on which it is based, has the properties that are attributed to it.

The process presented and applied in this paper will enhance the ability of the development team to express, assess, and validate the assumptions associated with the MLS LAN.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PROTOCOL SPECIFICATIONS

A. INTRODUCTION

The MLS LAN TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols are all presented as part of a proposed communications framework in the master's thesis by J. D. Wilson entitled: A Trusted Connection Framework For Multilevel Secure Local Area Network.[1] The specifications of each of the aforementioned protocols are presented in this chapter in the following format:

- Protocol Requirements
- Authorized MLS LAN Entities
- (For each Authorized Entity)
  - Authorized Messages
  - Authorized States and Transitions
  - Additional Information (If necessary)

The protocol requirements section gives the requirements of the particular protocol quoted directly from the authoritative work by J. D. Wilson.[1] The authorized participants section gives the MLS LAN entities that are authorized to engage in the protocol and any general restrictions on that use. The next sections are provided for each of the MLS LAN entities that are authorized to engage in the specific protocol and present the guidelines that the authorized participant must implement. These sections will cover the authorized messages, states and transitions, and any additional information that is relevant to the correct implementation of the protocol by the entity. The information in this chapter is based on the comprehensive information unit mapping, which is described and presented in appendix A. The specifications presented here are used as the basis for the development of the formal protocol specifications presented in chapter V.

B. TCB-TO-TCBE PROTOCOL

The TCB-to-TCBE protocol provides support for communications between a TCBE equipped workstation and the TCB Extension Server.[1] The TCBE equipped

workstation uses the protocol to gain secure attention from the TCB Extension Server. The TCB Extension Server uses the protocol to control the actions of the TCBE.

1. Requirements

The protocol attempts to fulfill the following requirements from the Multilevel Secure Local Area Network Project: Protocol High Level Analysis Document[1], Version 1 Section 3.2.

- TCB-to-TCBE Protocol shall only be initiated only through "secure attention" key from user.
- TCB-to-TCBE Protocol shall support the trusted path security related operations necessary to establish the initial session such as "login" and "user identification and authentication" or for any specified user operations that require a trusted path, such as "logout", "set session level", downgrade, change user password, etc.
- TCB-to-TCBE protocol shall allow establishment of a session only following activation by the user.
- TCB-to-TCBE protocol shall control the actions of the TCBE through the specific TCBE state commands.

2. Authorized Entities

Given the requirements placed on the protocol, there are only two MLS LAN entities that are authorized to employ the TCB-to-TCBE protocol; the TCB Extension Server and TCBE equipped workstations.

3. TCB Extension Server

a. Packets

The TCB Extension Server is only authorized to implement TCB-to-TCBE Protocol Command Packets that have the following format.

- TCB Identifier Header (32-bit) – Identifies the TCBE that created the packet.
- Version Number(4-bit) – present version is 1
- Response Type (4-bit) – allowed values {0,1,2}
  - 0 = No Response
  - 1 = Response with Echo
  - 2 = Response without Echo
- Command (4-bit) – allowed values {0,1,2,3,4,5,6}
  - 0 = NOOP
  - 1 = Run
  - 2 = New
  - 3 = PCC Update
  - 4 = Resume

- 5 = Logout
- 6 = Disconnect

- Payload length (8-bit) – length of Payload in 32-bit words
- Reserved (16-bit) – set to value of zero
- Payload (variable number of 32-bit words) – data sent to the TCBE

According to the Mealy diagrams[1] in Figure 2 and Figure 3 the TCB Extension Server may produce the following command packets:

TCBE	TCB Extension	Command Packet
NR (RUN)	NR (RUN)	{<TCB IH>, <V>, 0, 1, <PL>, <R>, <U>}
NR (NEW)	n/a	{<TCB IH>, <V>, 0, 2, <PL>, <R>, <U>}
NR (RESUME)	NR (RESUME)	{<TCB IH>, <V>, 0, 4, <PL>, <R>, <U>}
NR (LOGOUT)	NR (LOGOUT)	{<TCB IH>, <V>, 0, 5, <PL>, <R>, <U>}
NR (DISCONNECT)	NR (DISCONNECT)	{<TCB IH>, <V>, 0, 6, <PL>, <R>, <U>}
RE (NOOP) (Session)	RE (NOOP) (Session Information)	{<TCB IH>, <V>, 1, 0, <PL>, <R>, P}
RE (NOOP) (SL)	RE (NOOP) (Level Change Prompt)	{<TCB IH>, <V>, 1, 0, <PL>, <R>, P}
RE (NOOP) (SG)	RE (NOOP) (Group Change Prompt)	{<TCB IH>, <V>, 1, 0, <PL>, <R>, P}
RE (NOOP) (Username)	RE (NOOP) (Username Prompt)	{<TCB IH>, <V>, 1, 0, <PL>, <R>, P}
n/a	RE (NOOP) (User Interface Menu)	{<TCB IH>, <V>, 1, 0, <PL>, <R>, P}
RWOE (NOOP) (Password)	RWOE (NOOP) (Password Req)	{<TCB IH>, <V>, 2, 0, <PL>, <R>, P}
RWOE (NOOP) (UPDATE PCC)	RWOE (UPDATE PCC) or UPDATE PCC	{<TCB IH>, <V>, 2, 3, <PL>, <R>, P}

Table 1. Command Packet Information Presented in Figures 2 and 3.

b. States and Transitions

The TCB Extension Server's states are defined by five Boolean state variables: Power, Connect to TCBE, User Logged In, Session Operations, and Level Change. While there are 32 possible TCB Extension Server states, only six states are authorized.

State Number	Power	Connect to TCBE	User Logged In	Session Operations	Level Change	NAME
0	Off	No	No	No	No	Power Off
1	On	No	No	No	No	Idle
2	On	Yes	No	No	No	Connected
3	On	Yes	Yes	No	No	Logged in
4	On	Yes	Yes	Yes	No	Running
5	On	Yes	Yes	Yes	Yes	Trusted Session Processing

Table 2. Authorized TCB Extension Server States (From Ref 1)

There are a finite number of authorized transitions between the TCB Extension Server states presented in Table 2. These transitions are summarized in Table 3 and presented in their original form in Figure 2.

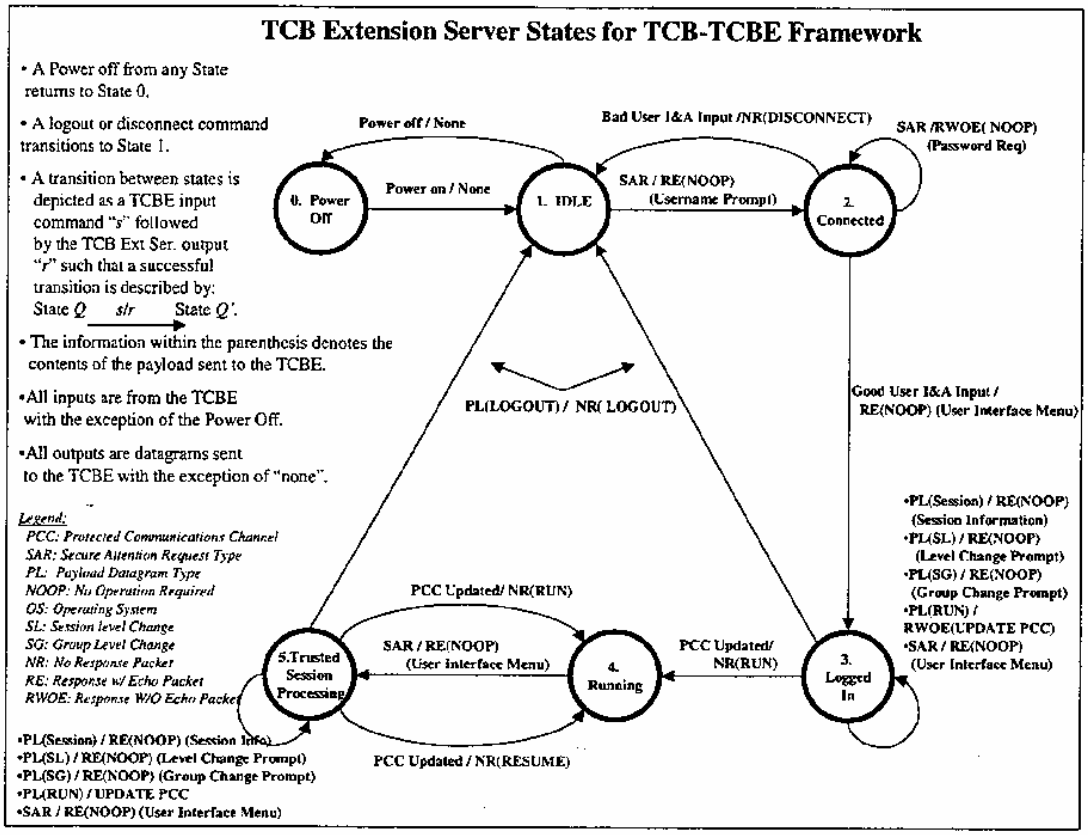


Figure 2. TCB Extension Server States for TCB-to-TCBE Framework (From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Idle [1]	SAR	RE(NOOP) (UserName Prompt)	Connected [2]
Connected [2]	SAR	RWOE (NOOP) (Password Req)	Connected [2]
Connected [2]	Bad User I&A Input	NR (DISCONNECT)	Idle [1]
Connected [2]	Good User I&A Input	RE (NOOP) (User Interface Menu)	Logged In [3]
Logged In [3]	PL (Session)	RE (NOOP) (Session Information)	Logged In [3]
Logged In [3]	PL (Session Level Change)	RE (NOOP) (Level Change Prompt)	Logged In [3]
Logged In [3]	PL (Group Level Change)	RE (NOOP) (Group Change Prompt)	Logged In [3]
Logged In [3]	PL (RUN)	RWOE (UPDATE PCC)	Logged In [3]
Logged In [3]	SAR	RE (NOOP) (User Interface Menu)	Logged In [3]
Logged In [3]	PCC Updated	NR (RUN)	Running [4]
Logged In [3]	PL (LOGOUT)	NR (LOGOUT)	Idle [1]
Running [4]	SAR	RE (NOOP) (User Interface Menu)	Trusted Session Processing [5]
Trusted Session Processing [5]	PCC Updated	NR (RUN)	Running [4]
Trusted Session Processing [5]	PCC Updated	NR (RESUME)	Running [4]
Trusted Session Processing [5]	PL (Session)	RE (NOOP) (Session Info)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL (Session Level Change)	RE (NOOP) (Level Change Prompt)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL (Group Level Change)	RE (NOOP) (Group Change Prompt)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL (RUN)	RWOE (UPDATE PCC)	Trusted Session Processing [5]
Trusted Session Processing [5]	SAR	RE (NOOP) (User Interface Menu)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL (LOGOUT)	NR (LOGOUT)	Idle [1]

Table 3. Summary of TCB Extension Server State Transitions from Figure 2.

4. TCBE Equipped Workstations

a. Packets

TCBE equipped workstations are only authorized to implement TCB-to-TCBE Protocol Payload Packets that have the following format.

- TCB Identifier Header (32-bit) – Identifies the TCBE that created the packet.
- Version Number (4-bit) – present version is 1
- Payload Type (4-bits) – allowed values {0,1,2}
  - 0 = Secure Attention Request
  - 1 = Response
  - 2 = PCC Updated
- Payload Length (8-bit) length of Payload in 32-bit words
- Reserved (16-bit) – set to value of zero
- Payload (variable number of 32-bit words) – data sent to the TCB Extension Server

According to the Mealy diagrams[1] in Figure 2 and Figure 3, a TCBE equipped workstation may produce the following Payload packets:

Figure 2	Figure 1	Response Packet
SAR	SAR	{<TCB IH>, <V>, 0, <PL>, <R>, <U>}
n/a	Payload Datagram Type (Session)	{<TCB IH>, <V>, 1, <PL>, <R>, <U>}
n/a	Payload Datagram Type (Session Level Change)	{<TCB IH>, <V>, 1, <PL>, <R>, <U>}
n/a	Payload Datagram Type (Session Group Change)	{<TCB IH>, <V>, 1, <PL>, <R>, <U>}
n/a	Payload Datagram Type (RUN)	{<TCB IH>, <V>, 1, <PL>, <R>, <U>}
n/a	Payload Datagram Type (LOGOUT)	{<TCB IH>, <V>, 1, <PL>, <R>, <U>}
n/a	PCC Updated	{<TCB IH>, <V>, 2, <PL>, <R>, <U>}

Table 4. Summary of Payload Packets Presented in Figures 2 and 3.

b. States and Transitions

The TCBE's states are defined by three Boolean state variables: Power, Trusted Path Operations, and Client OS Loaded. While there are eight possible TCB Extension Server states, only five states are authorized.

State Number	Power	Trusted Path Operations	Client OS Loaded	Name
0	Off	No	No	Power Off
1	On	No	No	Idle
2	On	No	Yes	Untrusted Operations
3	On	Yes	No	Trusted Processing
4	On	Yes	Yes	Trusted Session

Table 5. Authorized TCBE States (From Ref 1)

There are a finite number of authorized transitions between the TCBE states presented in Table 5. These transitions are summarized in Table 6 and presented in their original format in Figure 3.

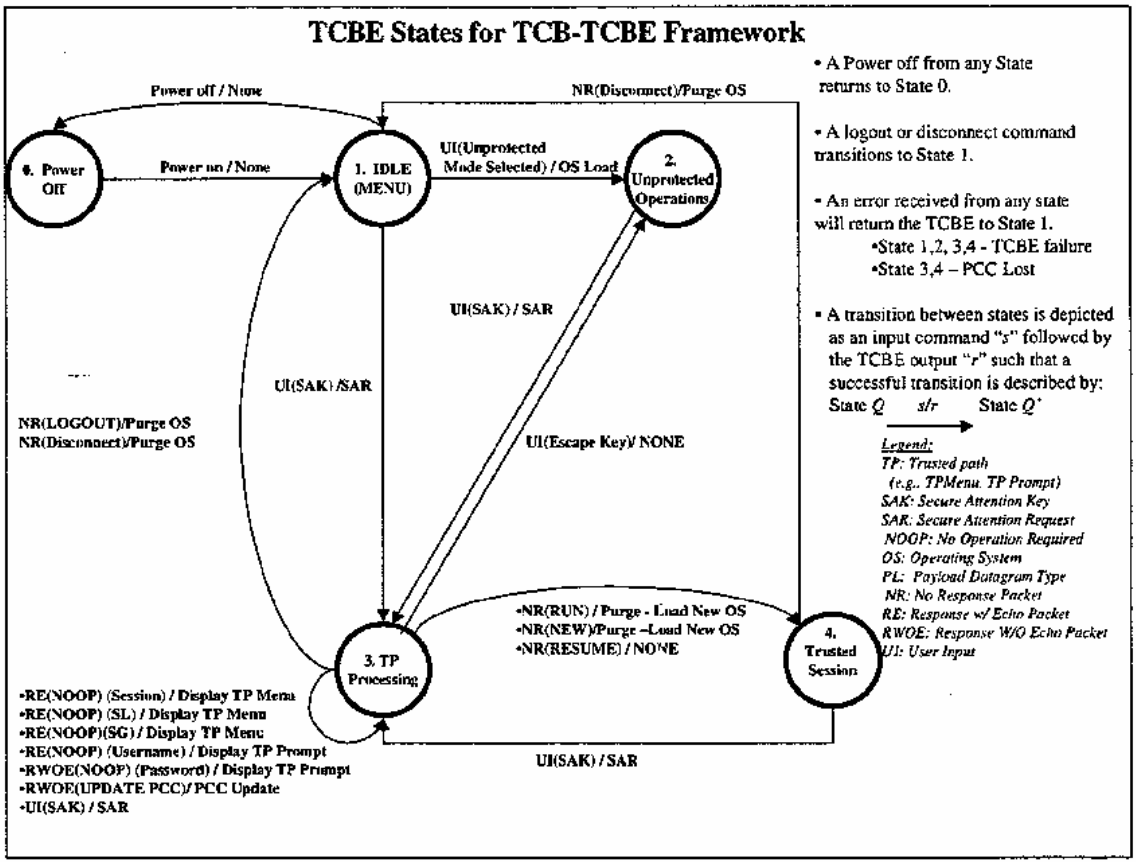


Figure 3. TCBE States for TCB-TCBE Framework (From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Idle [1]	UI (SAK)	none	TP Processing [3]
TP Processing [3]	RE (NOOP) (Session)	Display TP Menu	TP Processing [3]
TP Processing [3]	RE (NOOP) (SL)	Display TP Menu	TP Processing [3]
TP Processing [3]	RE (NOOP) (SG)	Display TP Menu	TP Processing [3]
TP Processing [3]	RE (NOOP) (Username)	Display TP Prompt	TP Processing [3]
TP Processing [3]	RWOE (NOOP) (Password)	Display TP Prompt	TP Processing [3]
TP Processing [3]	RWOE (UPDATE PCC)	Update PCC	TP Processing [3]
TP Processing [3]	UI (SAR)	none	TP Processing [3]
TP Processing [3]	NR (RUN)	Purge - Load OS	Trusted Session [4]
TP Processing [3]	NR (RESUME)	none	Trusted Session [4]
TP Processing [3]	NR(LOGOUT)	Purge OS	Idle [1]
TP Processing [3]	NR (Disconnect)	Purge OS	Idle [1]
Trusted Session [4]	UI (SAR)	none	TP Processing [3]
Trusted Session [4]	NR (Disconnect)	Purge OS	Idle [1]
FUTURE WORK			
Idle [1]	UI (Unprotected Mode Selected)	OS Load	Unprotected Operations [2]
Unprotected Operations [2]	UI (SAK)	none	TP Processing [3]
TP Processing [3]	NR (NEW)	Purge - Load OS	Trusted Session [4]
TP Processing [3]	UI (Escape Key)	none	Unprotected Operations [2]

Table 6. Summary of TCBE State Transitions Presented in Figure 3

C. SESSION STATUS PROTOCOL

The impetus for the Session Status Protocol is two fold. The first is the necessity for the TCB Extension Server to be able to create, modify, and delete entries in the Session Status Database. The second is the necessity for other MLS LAN entities to be able to acquire the session status values associated with a particular MLS LAN user.

1. Requirements

The protocol has following requirements from the Multilevel Secure Local Area Network Project's Project: Protocol High Level Analysis Document[1], Version 1 Section 3.3.

- The Session Status Protocol shall be initiated for every instantiation or modification of any information concerning the status of a user's current session.
- The Session Status Protocol shall support trusted communications between the TCB Extension Server and the Session Database Server, which is responsible for the maintenance of user-session security information.
- The Session Status Protocol shall support the encapsulation of session information, such as TCBE Identification Number, User Identification, Current Session Status, etc.

2. Authorized Entities

Given the requirements placed on the protocol, there are three MLS LAN entities authorized to employ the Session Status protocol: the TCB Extension Server, the Session Database Server, and Secure Session Servers.

3. TCB Extension Server

a. Packets

The TCB Extension Server is only authorized to implement Secure Session Protocol Request Packets, which have the following format:

- TCB Identifier Header (32-bit) – Identifies the TCBE that created the packet. (TCBE ID)
- User Session ID (32-bit) – TCBE ID
- Version Number (4-bit) – present version is 1
- Command (4-bits) – allowed values {0,1,2,3}
  - 0 = Create
  - 1 = Modify
  - 2 = List
  - 3 = Delete
- Payload Length (8-bit) length of Payload in 32-bit words

- Reserved (16-bit) – set to value of zero
- Payload (variable number of 32-bit words) –contains user and session information contained in attribute name / data as in:
  - User ID: <User ID>
  - Current Session Level: < Session level>
  - Current Integrity Level: <Integrity level>
  - Current Group Setting: <Group setting>
  - Running: <Boolean flag>

According to the Mealy diagram in Figure 4 the TCB Extension Server may produce the following Session Server Protocol Request packets:

Figure 4	Request Datagram
Request (CREATE)	{<TCB IH>, <UI>, <V>, 0, <PL>, <R>, <P>}
Request (MODIFY) (Trusted Path Processing Info)	{<TCB IH>, <UI>, <V>, 1, <PL>, <R>, <U>}
Request (LIST)	{<TCB IH>, <UI>, <V>, 2, <PL>, <R>, <U>}
Request (DELETE)	{<TCB IH>, <UI>, <V>, 3, <PL>, <R>, <U>}

Table 7. Session Status Protocol: TCB Extension Server Packets

b. States and Transitions

The Session Status protocol does not have states defined semantically within its own context but rather bases its states and transitions descriptions on subset of states established by the TCB-to-TCBE Protocol. The TCB Extension Server can send a List packet regardless of its internal state, but is only authorized to send Modify, Create, or Delete packets from TCB-to-TCBE Protocol states [2], state [3], and state [5].

State Number	Power	Connect to TCBE	User Logged In	Session Operations	Level Change	NAME
2	On	Yes	No	No	No	Connected
3	On	Yes	Yes	No	No	Logged in
5	On	Yes	Yes	Yes	Yes	Trusted Session Processing

Table 8. Session Server Protocol: Authorized TCB Extension Server States (Adapted From Ref 1)

There are a finite number of authorized transitions between the TCBE states presented in Table 8. These transitions are summarized in Table 9 and are presented in a Mealy diagram in Figure 4.

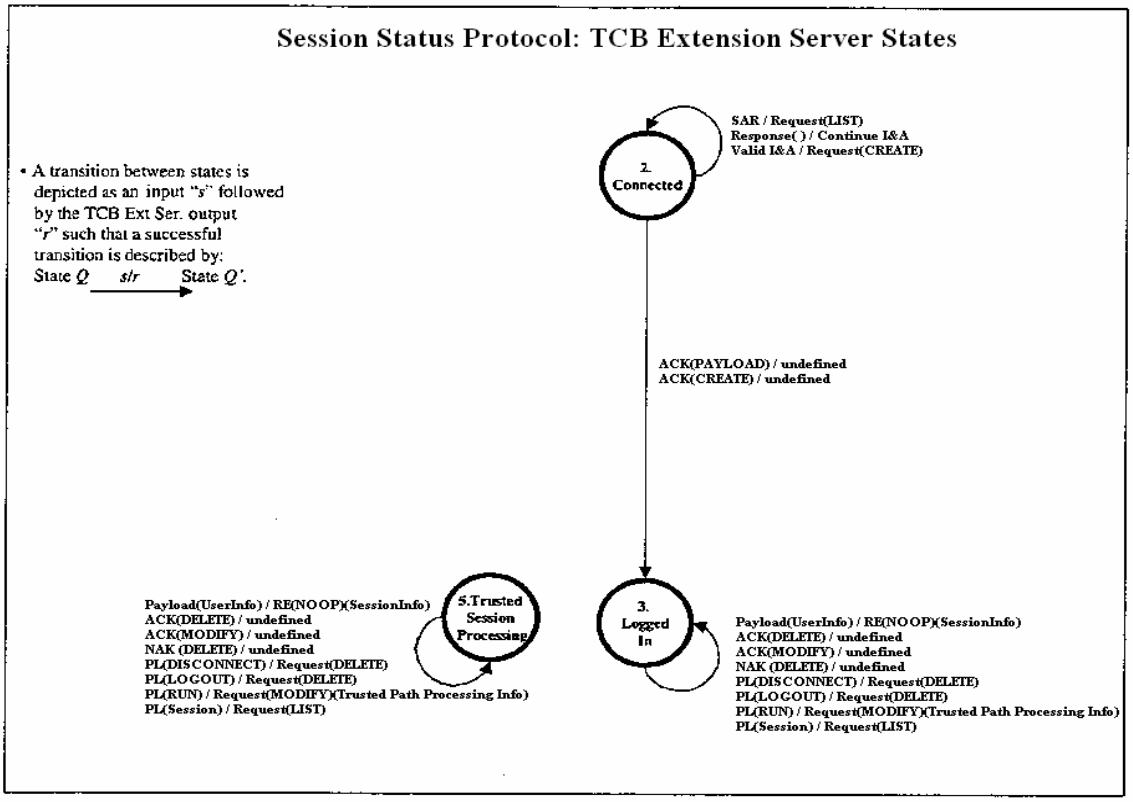


Figure 4. TCB Extension Server States for Session Status Protocol (Adapted From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Connected [2]	SAR	Request(LIST)	Connected [2]
Connected [2]	Response( )	Continue I&A	Connected [2]
Connected [2]	Valid I&A	Request(CREATE)	Connected [2]
Connected [2]	ACK(PAYLOAD)	undefined	Logged In [3]
Connected [2]	ACK(CREATE)	undefined	Logged In [3]
Logged In [3]	Payload(UserInfo)	RE(NOOP)(SessionInfo)	Logged In [3]
Logged In [3]	ACK(DELETE)	undefined	Logged In [3]
Logged In [3]	ACK(MODIFY)	undefined	Logged In [3]
Logged In [3]	NAK(DELETE)	undefined	Logged In [3]
Logged In [3]	PL(DISCONNECT)	Request(DELETE)	Logged In [3]
Logged In [3]	PL(LOGOUT)	Request(DELETE)	Logged In [3]
Logged In [3]	PL(RUN)	Request(MODIFY)(Trusted Path Processing Info)	Logged In [3]
Logged In [3]	PL(Session)	Request(LIST)	Logged In [3]
Trusted Session Processing [5]	Payload(UserInfo)	RE(NOOP)(SessionInfo)	Trusted Session Processing [5]
Trusted Session Processing [5]	ACK(DELETE)	undefined	Trusted Session Processing [5]
Trusted Session Processing [5]	ACK(MODIFY)	undefined	Trusted Session Processing [5]
Trusted Session Processing [5]	NAK(DELETE)	undefined	Trusted Session Processing [5]
Trusted Session Processing [5]	PL(DISCONNECT)	Request(DELETE)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL(LOGOUT)	Request(DELETE)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL(RUN)	Request(MODIFY)(Trusted Path Processing Info)	Trusted Session Processing [5]
Trusted Session Processing [5]	PL(Session)	Request(LIST)	Trusted Session Processing [5]

Table 9. Summary of TCBE State Transitions Presented in Figure 4.

4. Session Database Server

a. Packets

The Session Database Server is only authorized to implement Secure Session Protocol Response Packets.

- TCB Identifier Header (32-bit) – Identifies the TCBE that created the packet. (TCBE ID)
- User Session ID (32-bit) – TCBE ID
- Version Number (4-bit) – present version is 1
- Response (4-bits) – allowed values {0,1,2}
  - 0 = ACK Response
  - 1 = NAK Response
  - 2 = Payload Response
- Payload Length (8-bit) length of Payload in 32-bit words
- Reserved (16-bit) – set to value of zero
- Payload (variable number of 32-bit words) –contains user and session information contained in attribute name / data as in:
  - User ID: <User ID>
  - Current Session Level: < Session level>
  - Current Integrity Level: <Integrity level>
  - Current Group Setting: <Group setting>
  - Running: <Boolean flag>
  - Error: <reason for failure>

According to the Mealy diagram in Figure 5 the Session Database Server may produce the following Session Status protocol packets:

Figure 5	Response Datagram
ACK (CREATE)	{<TCB IH>, <TBC ID>, <V>, 0, <PL>, <R>, <P>}
NAK (CREATE)	{<TCB IH>, <TBC ID>, <V>, 1, <PL>, <R>, <P>}
ACK (MODIFY)	{<TCB IH>, <TBC ID>, <V>, 0, <PL>, <R>, <P>}
NAK (MODIFY)	{<TCB IH>, <TBC ID>, <V>, 1, <PL>, <R>, <P>}
Payload (Trusted Path Processing Info)	{<TCB IH>, <TBC ID>, <V>, 2, <PL>, <R>, <P>}
NAK (LIST)	{<TCB IH>, <TBC ID>, <V>, 1, <PL>, <R>, <P>}
ACK (DELETE)	{<TCB IH>, <TBC ID>, <V>, 0, <PL>, <R>, <P>}
NAK (DELETE)	{<TCB IH>, <TBC ID>, <V>, 1, <PL>, <R>, <P>}

Table 10. Session Database Server Response Packets

b. States and Transitions

The Session Status protocol does not have states defined semantically within its own context but rather bases its states and transitions descriptions on subset of states established by the TCB-to-TCBE Protocol. The Session Database Server is

assumed to have only a single state variable POWER. Once the Session Database server enters state [1] it is authorized receive request packets and to send response packets.

State Number	Power	Name
0	Off	Power Off
1	On	Idle

Table 11. Implicitly Authorized Session Database Server States

There are a finite number of authorized transitions between states of the Session Database Server. These states are presented in Table 11. The transitions are summarized in Table 12 and presented in the Mealy diagram in Figure 5.

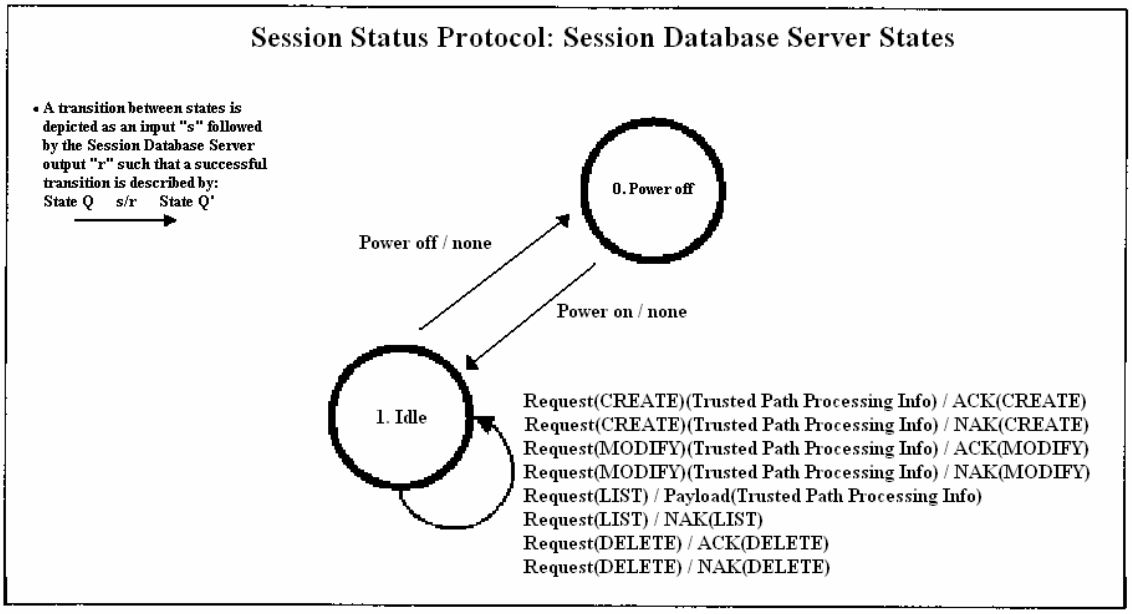


Figure 5. Session Database Server States for the Session Status Protocol (Adapted From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Idle [1]	Request(CREATE)(Trusted Path Processing Info)	ACK(CREATE)	Idle [1]
Idle [1]	Request(CREATE)(Trusted Path Processing Info)	NAK(CREATE)	Idle [1]
Idle [1]	Request(MODIFY)(Trusted Path Processing Info)	ACK(MODIFY)	Idle [1]
Idle [1]	Request(MODIFY)(Trusted Path Processing Info)	NAK(MODIFY)	Idle [1]
Idle [1]	Request(LIST)	Payload(Trusted Path Processing Info)	Idle [1]
Idle [1]	Request(LIST)	NAK(LIST)	Idle [1]
Idle [1]	Request(DELETE)	ACK(DELETE)	Idle [1]
Idle [1]	Request(DELETE)	NAK(DELETE)	Idle [1]

Table 12. Summary of State Transitions in Figure 5.

D. TCBE-TO-SESSION SERVER PROTOCOL

The TCBE-to-Session Server Protocol was developed in order to ensure that application layer protocols are only accessible to the appropriate users. It facilitates this by providing a way for TCBE equipped workstations to provide a unique identifier to a server that can establish "the proper session level connectivity to the appropriate MLS LAN Application Protocol Server"[1]

1. Requirements

The protocol has the following requirements from the Multilevel Secure Local Area Network Project's Project: Protocol High Level Analysis Document[1], Version 1 Section 3.4.

- The TCBC-to-Session Server Protocol shall only be initiated following the establishment of an authorized session between the client workstation and the TCB.
- The TCBC-to-Session Server Protocol shall support the encapsulation of information from the client workstation necessary for the identification and validation of the user's session sensitivity level and application service request.
- The TCBC-to-Session Server Protocol shall allow communications between a client and an MLS LAN Application Protocol Server only following positive validation of the user's session sensitivity level and the authorization for the specific application service.

2. Authorized Entities

Given the requirements placed on the protocol, there are two MLS LAN entities authorized to employ the TCBE-to-Session Server protocol; TCBE equipped workstations and Secure Session Servers.

3. TBCE Equipped Workstations

a. Packets

The TCBE equipped workstation is authorized to generate TCBE-to-Session Server Identification Packet.

- TCB Identifier Header (32-bit) – Identifies the TCBE that created the packet. (TCBE ID)
- TCBE Identification Number (32-bit) – Identifies the TCBE that created the packet (TCBE ID)
- Version Number (4-bit) – present version is 1
- Payload Length (8-bit) length of Payload in 32-bit words
- Reserved (20-bit) – set to value of zero
- Payload (variable number of 32-bit words) – this field is empty in this version of the protocol

According to the Mealy diagram in Figure 6 the TCBE may produce the following TCBE-to-Session Server packets.

Figure 6	Response Packet
Identification (TCBE)	{<TCB IH>, <TCB IH>, <V>, <PL>, <R>, <P>}

Table 13. Summary of Identification Packets Presented in Figure 6.

b. States and Transitions

There are no states defined specifically for the TCBE equipped workstations in the TCBE-to-Session Server Protocol. The protocol states referenced are based on the states established by the TCB-to-TCBE Protocol. TCBE equipped workstations are only authorized to send TCBE-to-Session Server Identification Packets in state [4].

State Number	Power	Trusted Path Operations	Client OS Loaded	Name
4	On	Yes	Yes	Trusted Session

Table 14. TCBE-to-Session Server: Authorized TCBE States

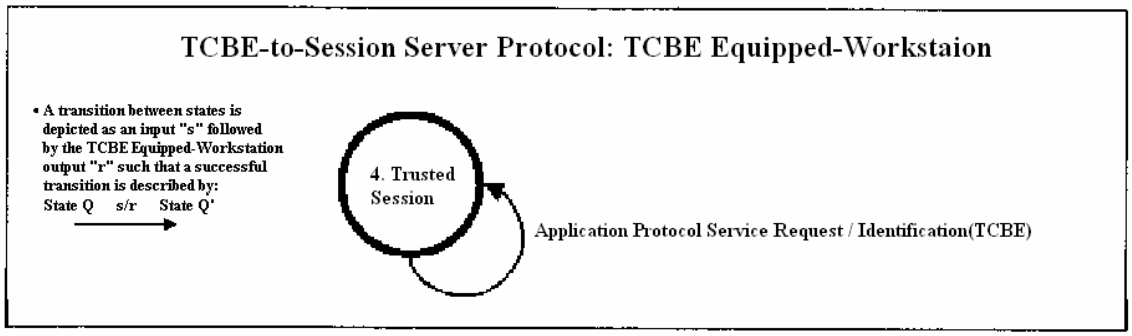


Figure 6. TCBE Equipped-Workstation States for TCBE-to-Session Server Protocol (Adapted From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Trusted Session [4]	Application Protocol Service Request	Identification(TCBE)	Trusted Session [4]

Table 15. Summary of State Transitions Presented in Figure 6.

4. Secure Session Servers

Secure Session Servers are responsible for protecting application layer protocols such as FTP and HTTP from unauthorized users. There is a one to one ratio of Secure Session Servers to higher layer protocols in the MLS LAN. The server is responsible for validating that the user has established a session with the TCB and that the user has the appropriate sensitivity and integrity setting to access the application protocol.

a. Packets

The Secure Session Server is not authorized to produce TCBE-to-Session Server Protocol Packets. It is only authorized to receive TCBE-to-Session Server Protocol Packets from TCBE equipped workstations.

b. States and Transitions

The Secure Session Server is authorized to accept TCBE-to-Session Server Protocol Packets in state [1]. There is only one state Boolean variable presented for the Secure Session Server: Power.

State Number	Power	Name
0	Off	Power Off
1	On	Idle

Table 16. Implicitly Authorized Secure Session Server States

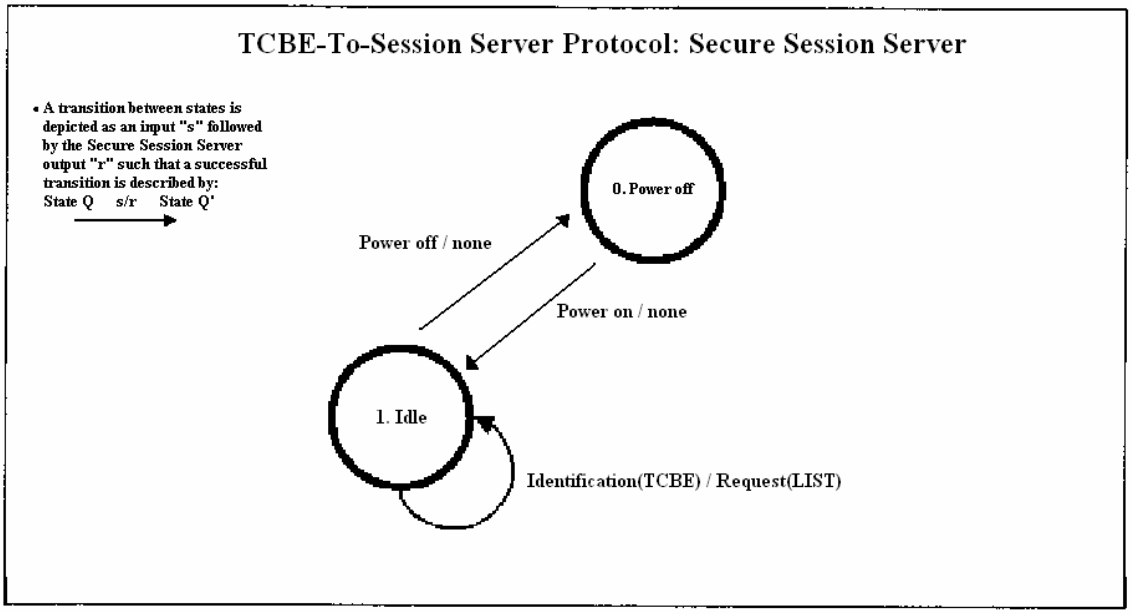


Figure 7. Secure Session Server States for the TCBE-to-Session Server Protocol (Adapted From Ref 1)

START STATE	INPUT	OUTPUT	END STATE
Idle [1]	Identification(TCBE)	Request(LIST)	Idle [1]

Table 17. Summary of State Transitions Presented in Figure 7.

E. SUMMARY OF SPECIFICATIONS

The protocols presented previously interact to form a framework that enables the components of the MLS LAN to securely interact. An example of that framework is presented in Figure 8. Figure 8 uses four different colors to add meaning to various interactions. The two-headed blue arrows represent PCC establishment between two

MLS LAN entities. The color red emphasizes areas where assumptions were made about protocol interactions. Blue represents actions or processing internal to the particular MLS LAN entity, and the green two-headed arrow represents a connection between an authenticated user and an application protocol server.

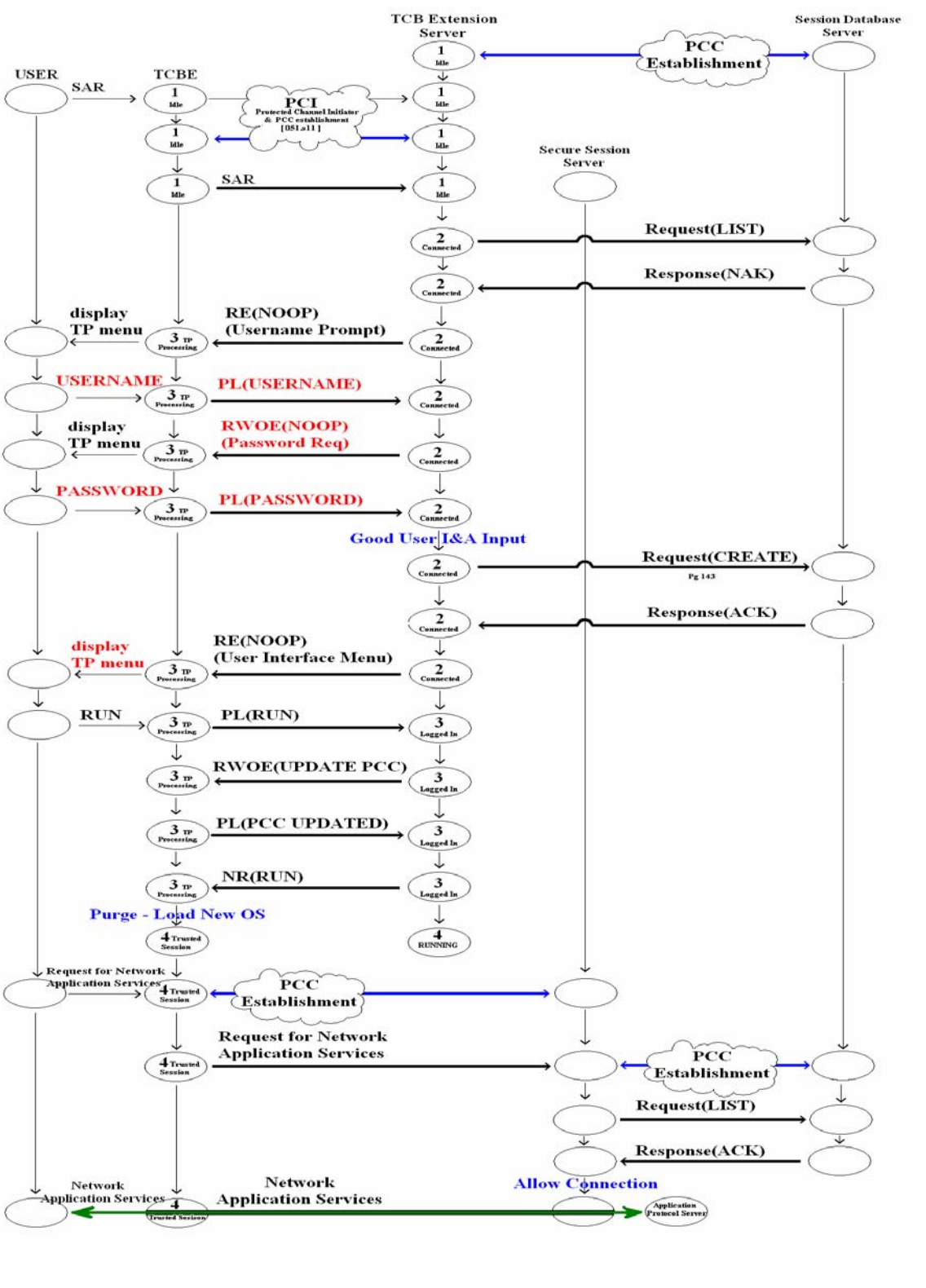


Figure 8. Protocol Interaction of the MLS LAN

V. FORMAL PROPERTIES

There are many techniques used in formal protocol analysis. Each of the methods has both strengths and weaknesses. Many of the most widely used methods are presented in the background chapter of this paper. The method chosen for this paper is Strand Spaces, which was developed by F. Javier Thayer Fabrega, Jonathan Herzog, and Joshua Guttman.[30] This chapter will give a general explanation of Strand Spaces, followed by the formal properties of the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols expressed in Strand Space notation and presented as they relate to each entity of the network. The actual conversion of the informal protocol descriptions to Strand Space representations is presented in appendix B.

A. STRAND SPACES

Strand Spaces is similar to model checking, while at the same time incorporating the ability to use induction methods as well as presenting a very intuitive graphical representation of protocols. This graphical approach is "used as a heuristic for stating and proving correctness results."[31]

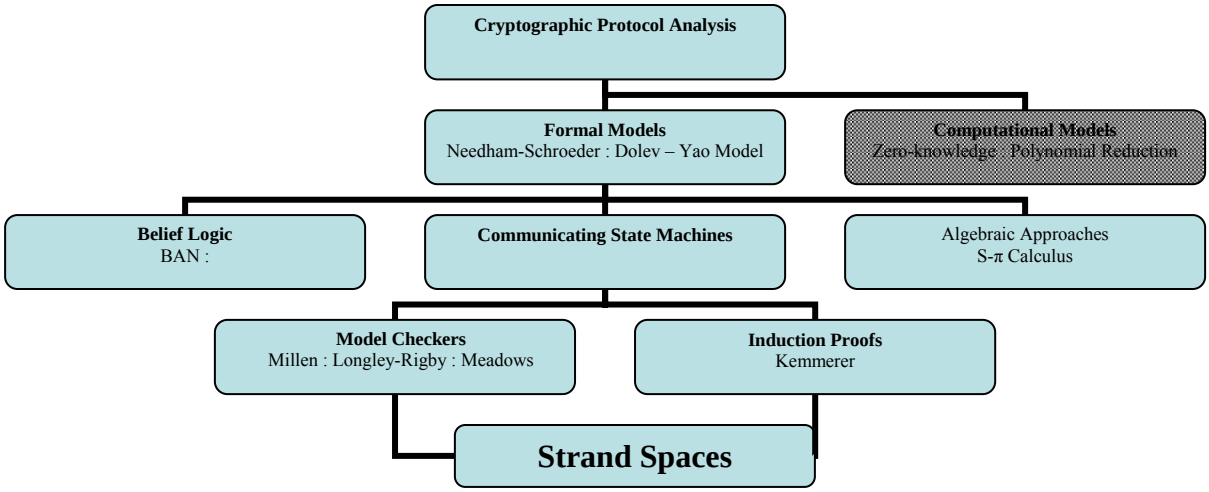


Figure 9. Strand Spaces' Relation to Other Developments in Formal Protocol Analysis (Adapted From Ref 6, 7, 8)

Strand Spaces have several advantages. They allow proofs that are simple, informative, and easily developed by hand. The formalisms easily handle assumptions that are impossible for some other formal analysis methods, for example assumptions about freshness of nonces and session keys. Another advantage is that it provides an explicit model of the intruder.

A full description of Stand Space formalisms is presented elsewhere[30,32,33,34] and readers are urged to consult those papers for complete coverage of the topic. This chapter only presents enough of a general description of Strand Space formalisms to make the notation used in the following protocol descriptions understandable.

There are seven concepts that are critical to the understanding of Strand Space formalisms. Those items are presented in the following order; ***terms***, ***strands***, ***bundles***, ***authorized participants***, ***secrecy***, ***freshness***, and the ***penetrator model***.[1]

1. Terms

An important part of any protocol is the information that participants pass between each other. In Strand Space formalisms these messages are referred to as ***terms***. ***Terms*** have a sub-term relationship defined. This means that a term can be made of a collection other terms. Protocols define which participants should send a specific ***term*** and which participants should receive ***terms***. This is reflected in Strand Spaces by creating an element called a "signed" ***term***. The new ***term*** is actually a tuple consisting of either a negative sign if the participant receives the original message or a positive sign if the participant sends the original term. These pairs can be represented by the form $<\sigma, \underline{\boldsymbol{\alpha}}>$ where σ is an element of the set {-, +} and $\underline{\boldsymbol{\alpha}}$ is an element of the set of all valid protocol messages. A few simple examples of ***terms*** are given in Figure 10.

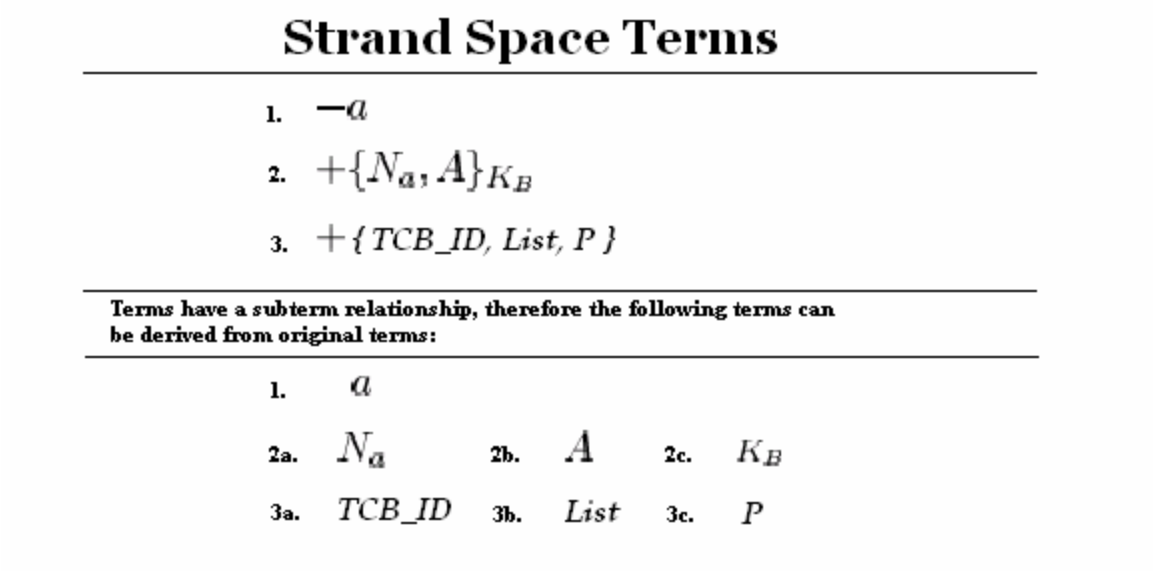


Figure 10. Simple Examples Strand Space Terms

2. Strands

A ***strand*** is a sequence of signed terms for a particular participant. A few examples of ***strands*** are given in Figure 11.

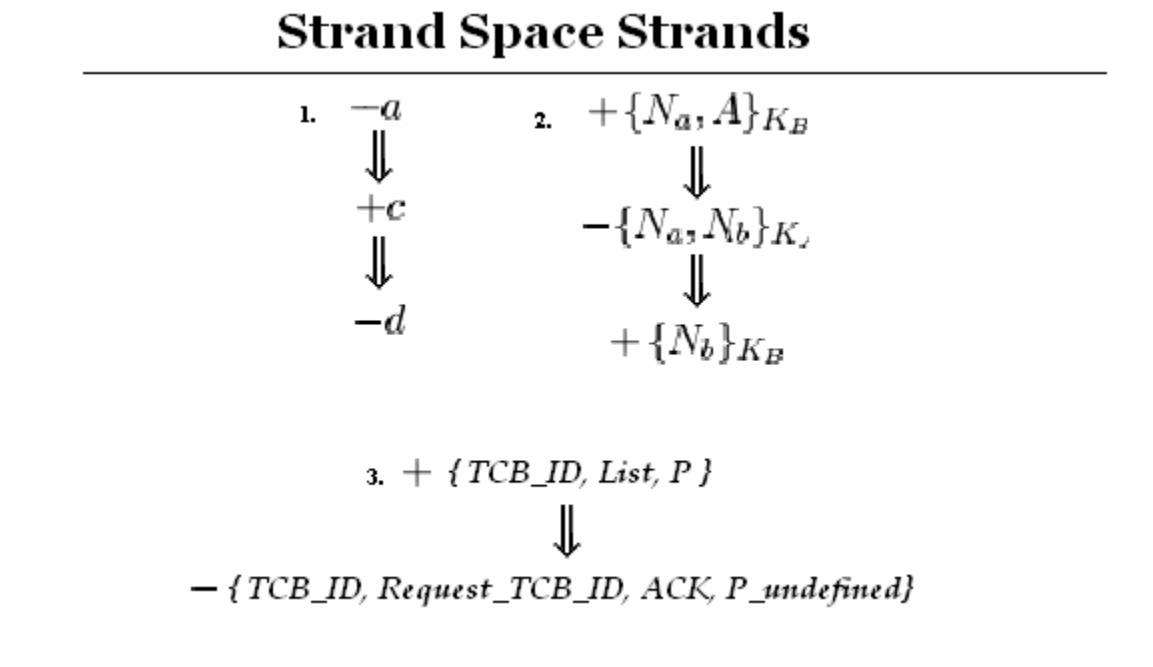


Figure 11. Simple Examples of Strand Space Strands

A ***strand*** is meant to represent a particular run of the protocol for a particular participant, "with specific values for all data items such as keys and nonces".[30] Connecting signed ***terms*** creates a ***strand***. Each ***strand*** has a linear progression starting with the first ***term*** and continuing one ***term*** at a time until the final ***term***. ***Strands*** can therefore be thought of as numbered sequence of signed ***terms***, indexed 1 through N. The connection between two ***terms*** in a ***strand*** is represented by the => symbol, normally written vertically. If n_1 and n_2 are both signed ***terms*** then n_1 => n_2 means that n_1's index number = n_2's index number -1.[30]

3. Bundles

A ***bundle*** is two or more "connected" ***strands***. ***Bundles*** are constructed by connecting a positively signed ***term*** from one participant to the equivalent negatively signed ***term*** of another participant. These connections are represented with a single arrow written between the two ***terms***. Therefore if both n_1 and n_2 are ***terms*** from different ***strands*** then $n_1 \rightarrow n_2$ implies that n_1 has a positive sign, n_2 has a negative sign, and that the unsigned ***terms*** of n_1 and n_2 are equal. An example is given in Figure 12.

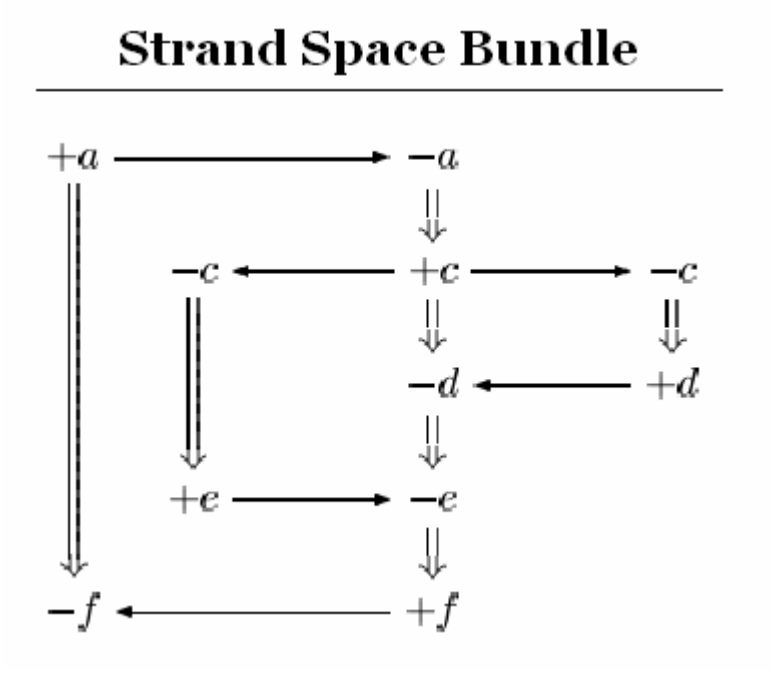


Figure 12. Simple Example #1 of Stand Space Bundle (Adapted From Ref 32)

There is another equivalent representation that incorporates participant names and a single ***term*** written above the ***bundle*** arrow representation. The ***term*** is understood to have a positive sign in the originating ***strand*** and a negative sign in the receiving ***strand***. An example of this notation is given in Figure 13.

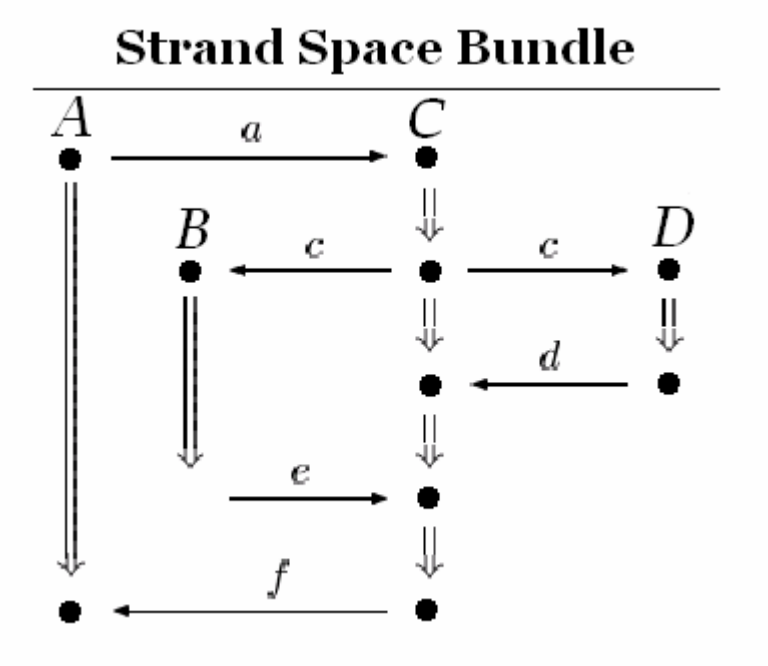


Figure 13. Simple Example #2 of Strand Space Bundle (Adapted From Ref 32)

This is the notation that is used in the graphical representations of protocol ***bundles*** in this paper.

4. Authorized Participants

Authorized participants are another important aspect of any protocol. Strand Spaces has a clear and formal definition of the items that must be defined for these participants. Each participant has a set of known information and a set of operations that it may perform. The set of known items may consist of other participant's public keys, established symmetric keys; the participant's own private key, and any other initially known pieces of information or previously acquired knowledge. The actions of each ***authorized participant*** are defined in the protocol definition. These authorized operations entail all the actions necessary to fulfill the participant's role in a successful run of the protocol. These may include the ability to encrypt a message using a known

key, decrypt a message using a known key, create packets of a curtain form, etc. Both the initially known items and the authorized operations are presented for each of the authorized participants.

5. Secrecy

Two properties that ***authorized participants*** must contend with in many protocols are ***secrecy*** and ***freshness***. In Strand Space representations the idea of ***secrecy*** is directly related to the ***terms*** that are sent between participants. A piece of information is considered secret if two principles hold:

- Authorized participants never send the piece of information
- Penetrator can not derive the secret from ***terms*** that are sent

6. Freshness

Freshness of a nonce or a timestamp is modeled efficiently in Strand Spaces. Only the originating participant can send the original ***term*** that contains the ***freshness*** item. Other participants may use this ***term*** within their normal set of operations but not before they have received it, thus enforcing the ***freshness*** property.

7. Penetrator Model

Stand Spaces has a well defined ***penetrator model***. The penetrator has the same two aspects as authorized participants; a set of initially known pieces of information and a set of actions that the penetrator can use to manipulate the information it knows. The Strand Space model of the penetrator follows the model set forth by Dolev and Yao.[12] This model gives the penetrator the ability to create, modify, and destroy any message on the network as long as the messages that are modified or created are possible using the known pieces of information and the actions that the penetrator can perform on that information.

VI. ANALYSIS OF RESULTS

The result of the analysis of the TCB-to-TCBE, Session Status, and TCBE-to-Session Status protocols is presented in three sub-sections. The first section, entitled Informal Protocol Description, presents areas that resulted in assumptions about the information relevant to the MLS LAN protocols as well as areas of particular interest. This section is supported by the work presented in appendix A. The second section, entitled Formal Protocol Description, suggests areas of interest that arose as result of the creation and hand evaluation of the formal Strand Space protocol representations. This section is based on the information presented in appendix B. The third section, entitled Automated Tool, presents the areas addressed and the results of an analysis using Millen's Constraint Analyzer.[8,37,38,39] This section is based on the material presented in appendix C. The three sections follow.

A. INFORMAL PROTOCOL DESCRIPTION

This section presents areas that resulted in assumptions about information pertaining to the protocol specifications as well as protocol areas of interest. This section is organized into seven sub-sections. The first three sub-sections entitled Terminology, Typographical, and Multiple Interpretations cover areas that resulted in assumptions about the meaning and intent of the information presented. The final four sub-sections, entitled Error Handling and Undefined Interactions, Loss of the TCB-to-TCBE Protocol Channel, Secure Session Database RUNNING Flag, and Extraneous Abilities present protocol areas that of interest and how this analysis addresses those areas.

1. Assumptions about Protocol Information

There is a tremendous amount of information presented on the MLS LAN and the protocols associated with it. Several assumptions about the meaning and intent of the information are made. These assumptions follow.

a. Terminology

Several naming conventions were used in the documentation. This resulted in a series of assumptions about name equivalency. These assumptions ranged from the quite obvious such as the equivalency of TCB-to-TCBE Protocol, TCB-TCBE

Protocol, and TCB-TCBE Connection Protocol, to the more difficult assumptions such as RE(NOOP)(SL) is equivalent to RE(NOOP)(Level Change Prompt). While these assumptions are relatively easy to assign correctly, the use of differing conventions can lead to confusion. Every attempt was made to identify all different but equivalent naming conventions; based on the information provided. This type of assumption could be minimized if official names and representations for all of the entities and packet representations were standardized for the entire project.

b. Typographical

A small number of questions about the meaning of the typography arose. We made several assumptions based on the relevant information. For example a reference[1] on page 141 to section 4.4.1.g is assumed to be 4.4.1.c. This assumption is based on the fact that no section 4.4.1.g is included in the document and that the content of section 4.4.1.c addresses relevant information to the section that contained the reference. Another example of typographical assumptions is the fact that the body of the document uses a numbering scheme for the states of the TCBE that is different from the numbering scheme presented in MLS LAN Connection framework. After careful analysis the numbering systems were determined to be equivalent and therefore the analysis uses the numbering system presented in the MLS LAN Connection Framework documentation exclusively.

c. Multiple Interpretations

The PCC protocol was not a focus of this analysis. However, it does illustrate a good example of possible multiple interpretations of information. The presentation of the PCC is based on an implementation of IPSec and its implementation in the MLS LAN is presented in Figure 14.

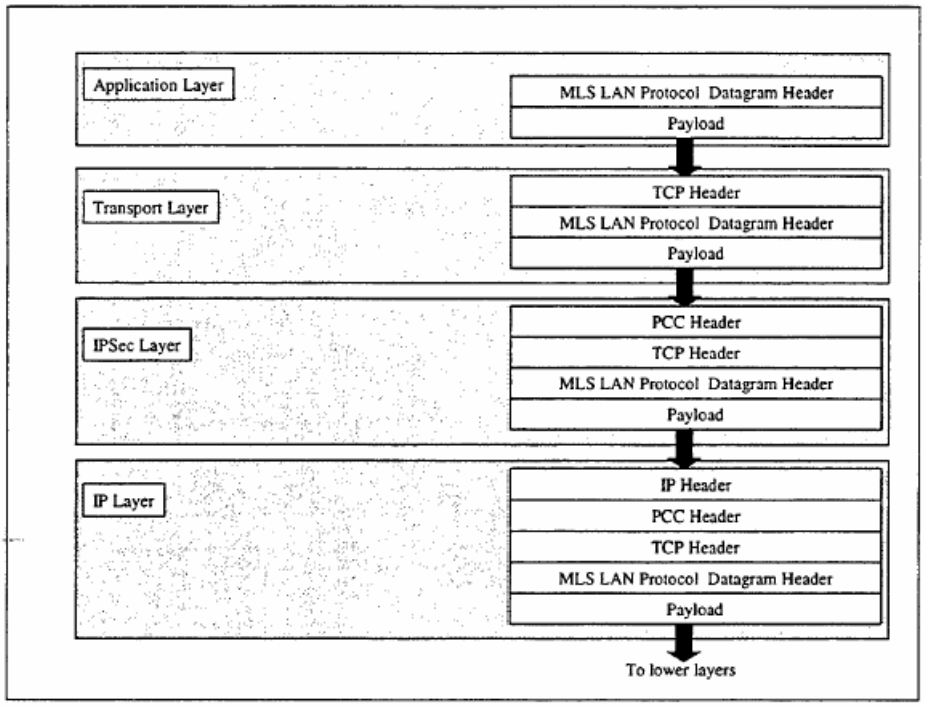


Figure 14. Originally entitled: MLS LAN Protocol Datagram Packaging (From Ref 1)

The documentation states that the "MLS LAN implement[s] IPSec in a BITS configuration and create[s] the Protected Communications Initiator".[1] The documentation also states that the BITS configuration of IPSec places the IPSec implementation in between the native IP layer and the network drivers. However, Figure 14 could be construed to mean that the IP Layer is not included in the IPSec header. If the PCC is implemented in the manner presently indicated in Figure 14 it would not satisfy the requirement of mutual "two-way" hardware identification presented in section 3.1.1.[1] This could be addressed by clearly showing that the IP layer information is contained in the PCC packet, if this is in fact the case. However, it should be noted that the implementation of the PCC was not evaluated in this paper. The properties that the PCC establishes according to the documentation are assumed to be fulfilled. This has two benefits. It allows analysis to focus on the three protocols developed by the MLS LAN development team that depend on the PCC. In addition, it allows the manner in which the PCC provides these properties to be evaluated, modified, and changed without affecting the validity of the analysis done here, as long as the properties of the PCC

remain intact. Once the PCC implementation has been finalized, it could be formally evaluated to prove that it provides the properties that the protocols evaluated in this paper depend on. Some work has already been done in this area with IPSec.[35,36]

2. Protocol Areas of Interest

The following areas are of particular interest as a result of the gathering of information and the construction of an informal protocol description.

a. Error Handling and Undefined Interactions

The error handling expected of the system is not explicitly stated in the documentation.[1] Descriptions of the mechanisms and expected consequences of error handling are also absent. The ability of a system to handle errors without entering a state that compromises the system is extremely important. However, for the purpose of this analysis these mechanisms were assumed to function properly.

There are several occasions in the protocol specifications where the description contains the phase: participants "will enter an interactive exchange."[1] A more detailed specification about the contents of the Payload section of both Command and the Payload packets used in this exchange would facilitate a more in depth formal analysis of this aspect of the protocol. This applies to both the session level negotiation and the group negotiation provided by the TCB-to-TCBE protocol. The interchange between the TCBE and the TCB Extension Server that constitutes the "User I&A" presented in Figure 2 on page 128 of the documentation[1] is not defined. Assumptions were made about the "interactive exchanges" in order to complete the analysis.

b. Loss of the TCB-to-TCBE Protocol Channel

Section 3.2.1.3 on page 88, in the Systems Requirements Document, states: "Once the session has been established, the TCB shall not allow the TCB-to-TCBE Protocol Channel to be broken without loss of network functionality with respect to shared resources, protocol services and applications provided by the MLS LAN".[1] The mechanisms that enable the TCB to enforce this requirement are not presented. In order to complete an analysis of the protocols, these properties and the enforcement mechanisms were assumed to function properly.

c. Secure Session Database RUNNING Flag

The entries in the Secure Session Database contain the following fields: USER ID; CURRENT SESSION LEVEL; CURRENT INTEGRITY LEVEL; CURRENT GROUP SETTING; RUNNING. The field entitled RUNNING is a flag that represents whether a user has started a current session or not. The TCB Extension Server uses the Request packet format to change the flag's setting. However, the interactions that change the RUNNING flag and the mechanisms employed to insure its validity are not explicitly presented. The documentation implicitly states that the absence of an entry in the Secure Session Database implies a user is "logged out"; with respect to a particular TCBE, and that the presence of a Secure Session Database entry implies the user is "logged in". Based on this information the RUNNING flag is assumed to be correctly modified when the TCB Extension Server sends a Session Status Protocol Request packet other that the LIST request.

d. Extraneous Abilities

"All TCB Entities may use the Request datagram to make query (LIST) requests of the Session Database Server."[1] If a TCBE is allowed to directly query the Session Database Server, a user might receive information about other users and their current settings. Therefore, the ability of TCBE equipped workstations to directly query the Session Database Server should be explicitly denied. This could be accomplish with the addition of the following sentences:

- The TCBE is not allowed to make query (LIST) requests of the Session Database Servers.
- The TCBE will be responsible for enforcement of this property.

The analysis to this point, based on the protocol information[1] and the assumptions developed from that information, has shown no major issues in the MLS LAN protocol framework or design.

B. FORMAL PROTOCOL DESCRIPTION

This section presents both assumptions about protocol information and interesting areas that were highlighted as a result of the creation and hand evaluation of the formal Strand Space protocol representations. This section is based on the information presented in appendix B.

1. Assumptions about Protocol Information

a. PCC

This paper presents an analysis of the TCB-to-TCBE, the Session Status, and the TCBE-to-Session Server protocols. However, because all three protocols depend on the Protected Communications Channel (PCC) to establish "a secure interaction communications channel"[1] and to enforce "the mutual authentication between two TCB entities"[1], the assumptions about the PCC and how these assumptions are modeled in the Strand Space representation needs to be explicitly stated. Figure 24 from appendix B gives a Strand Space bundle of a successful run of the protocols. However, that Figure does not incorporate a representation of the PCC. Since the present suggested implementation of the PCC is a version of IPSec the establishment of the PCC is treated as follows: Each pair of MLS LAN entities that establish a PCC channel during a single run of the protocols are assumed to have the symmetric keys necessary to implement that channel in their set of initially known items. The notation used in the Strand Space formalisms to represent the PCC is given in Figure 15.

PCC Strand Space Representation

General Form: { {PCC Dependent Protocol Packet}, <Nonce Associated with PCC>} Symmetric Key **Entities that Share the Key**

Example: { {TCB_ID, SAR, P_undefined}, N } K_{BC}

Shorthand Representation (Equivalent to Example, Used in Figures): {TCB_ID, SAR, P_undefined} K_{BC}

Figure 15. PCC Strand Space Representation

b. Version Numbering

A version number is included in each of the analyzed protocols. Presently there is only a single version of each protocol, for that reason the version number for each protocol is set to one. The version number is not included in any of the Strand Space formalisms constructed in appendix B. If different versions of the protocols are not

expected to interact then the version number information contained in the protocol packets is extraneous. If different versions of the protocols are expected to interact, this interaction and how these differences affect the assumptions of earlier implementations should be addressed as the new versions of the protocols are developed. Different protocol versions, by definition, are different in some manner from previous versions and how the different versions of the MLS LAN protocols interact could have a profound effect on the security properties of the network.

2. Areas of Interest

a. User I&A

The present protocol specification does not explicitly define what constitutes the payload section of a TCB-to-TCBE protocol SAR packet. The specification simply states that the variable length payload field "contains the data to be sent to the TCB Extension Server, typically, this will be the input from the user."[1] If the user name is not included in the TCB-to-TCBE protocol SAR packet a user could possibly again access to another users session, see Figure 16. Figure 16 also assumes that the time between the logout of user A and user F being allowed to connect to a network application server using user A's settings is less than the time needed by the mechanism that detects PCC lost. It also assumes that there is no mechanism within the TCB Extension Server, which changes the interaction between the TCB Extension Server and the Session Database Server when two different PCCs are established from the same TCBE. These are not trivial assumptions and protection mechanisms already in place may make the assumptions stated earlier impossible. However, future formal analysis efforts might benefit from additional detail regarding the payload field contents of the MLS LAN protocols.

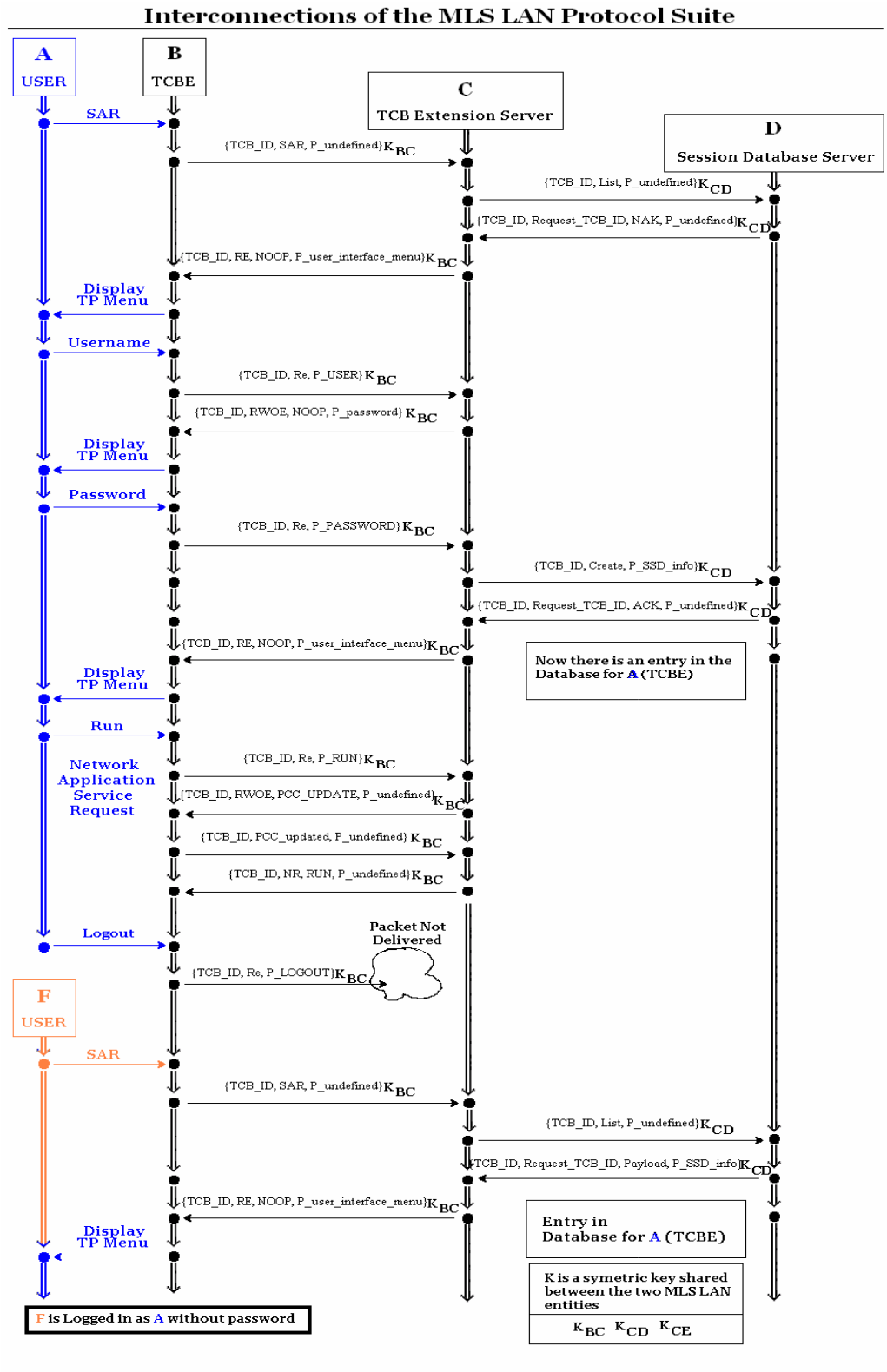


Figure 16. Interconnections of the MLS LAN Protocol Suite

b. TCB Extension Server – Session Database Server Connection

The loss of communications between the TCB Extension Server and the Session Database Server might allow unwarranted access to the MLS LAN.[1] This issue is presented in the protocol specification. This is an important aspect of the MLS LAN security framework that it will have an enormous effect on the security properties of the network as a whole.

3. Constraint Checker

The Constraint Checker is a tool developed by John Millen. Information pertaining to the Constraint Checker and the process used to arrive at the following results is presented in appendix C.

a. Results

The results from the modified protocol run are just as expected. They do not demonstrate any secrecy issues related to the tested terms from the protocols. While these results are promising for the secrecy properties of the MLS LAN as a whole, there are several important items to note about the testing. Authentication properties have not been included in this section of the analysis. The assumptions about the PCC may not accurately represent the future or even present PCC implementation. The analysis was limited to the interaction between a single TCBE, a single TCB Extension Server, a single Secure Database Server, and a single penetrator.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS

This paper presented a formal protocol analysis process and the results of applying that process to the MLS LAN: TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols. The results of the analysis completed at each of the steps in the process were presented in chapter six.

The first step in the process, took the information presented in the original documentation and formed an informal protocol specification of the three analyzed protocols. This step highlighted protocol requirements as well as the MLS LAN entities and the messages, states, and transitions associated with the protocol. The analysis completed during this process did not discover any major issues with the analyzed protocols. It does present several assumptions about the meaning and intent of the information used in the analysis. The assumptions were grouped into three general areas: terminology, typographical, and multiple interpretations. Additionally the analysis of the informal protocol specification suggested areas that might benefit from additional specification detail. These areas included: error handling and undefined interactions, loss of the TCB-to-TCBE protocol channel, the Secure Session Database RUNNING flag, and a possibly extraneous ability of one protocol participant.

The second step in the process built on the assumptions, specifications, and analysis completed in the first step and presented the items that the creation and hand evaluation of the formal Strand Space representations highlighted. Assumptions made at this stage of the analysis are presented in two general areas: those pertaining to the Protected Communications Channel (PCC) and those pertaining to protocol version numbering. Additionally the analysis of the Strand Space representations suggested one area of interest: payload field specification detail.

The final step in the process built on the assumptions, specifications, and formalisms completed in the previous steps of the process. It transformed the Strand Space protocol representations into an equivalent prolog based representation, which allowed a secrecy property of the three MLS LAN protocols were analyzed under an

limited set of conditions using Dr. John Millen's Constraint Checker. No secrecy issues were uncovered in this area of the analysis.

VIII. FUTURE WORK

There are several ways in which the work presented in this paper could be continued. The recommendations for future work fall into four general categories: expanding the coverage of items within the current assumption framework, addressing assumptions of the analysis, expanding the scope of the analysis, and creating a mapping from the protocol requirements to the protocol specifications. The rest of this section will suggest future work in these areas.

A. EXPAND COVERAGE WITHIN ASSUMPTION FRAMEWORK

This paper presented an analysis that is based on a set of assumptions. Future work could build on that set of assumptions and expand the properties of the MLS LAN TCB-TCBE Connection, Session Status, and TCBE-to-Session Server Connection protocols analyzed. The automated tool presented in appendix C was used to analyze security properties of a limited set of participants. The set of participants could be expanded, which would increase the confidence in the ability of the protocols to satisfy security properties of the network.

A natural extension of the specifications presented in appendix C would be to incorporate authentication properties, which could be analyzed with the help of Constraint Checker.[8,38,40]

B. ADDRESS ASSUMPTIONS OF THE ANALYSIS

Future work could attempt to reduce the set of assumptions used in this analysis. This process could evaluate reasons assumptions were necessary and collect additional information to alleviate the need for those assumptions. General areas that might benefit from this type of investigation follow:

- Protocol Specifications.
- Naming Conventions.
- PCC Properties.
- Error Handling.
- Participant Interactions
- Enforcement Mechanisms.

C. EXPAND SCOPE OF ANALYSIS

Future work could incorporate additional protocols and MLS LAN properties in the analysis. A natural addition would be to incorporate the Protected Communications Channel protocol in the analysis. Once the implementation of the PCC is finalized, its addition to the analysis would increase the confidence in the MLS LAN as a whole.

D. MAPPING PROTOCOL REQUIREMENTS TO SPECIFICATIONS

Future work could provide a mapping between system requirements and system specifications. This would provide a binding between these two levels of abstraction, which would enhance the ability to prove the system is a manifestation of the requirements.[41]

These are only a few of the possible future directions for this type of analysis. Each will have its own perils and rewards.

APPENDIX A: MAPPING

The primary objective of the thesis that this document supports is to formally analyze the MLS LAN TCB-TCBE Connection, Session Status, and TCBE-to-Session Server Connection protocols as they are presented by J. D. Wilson in his Master's Thesis: A Trusted Connection Framework for Multilevel Secure Local Area Network. Formally analyzing a protocol requires several steps.

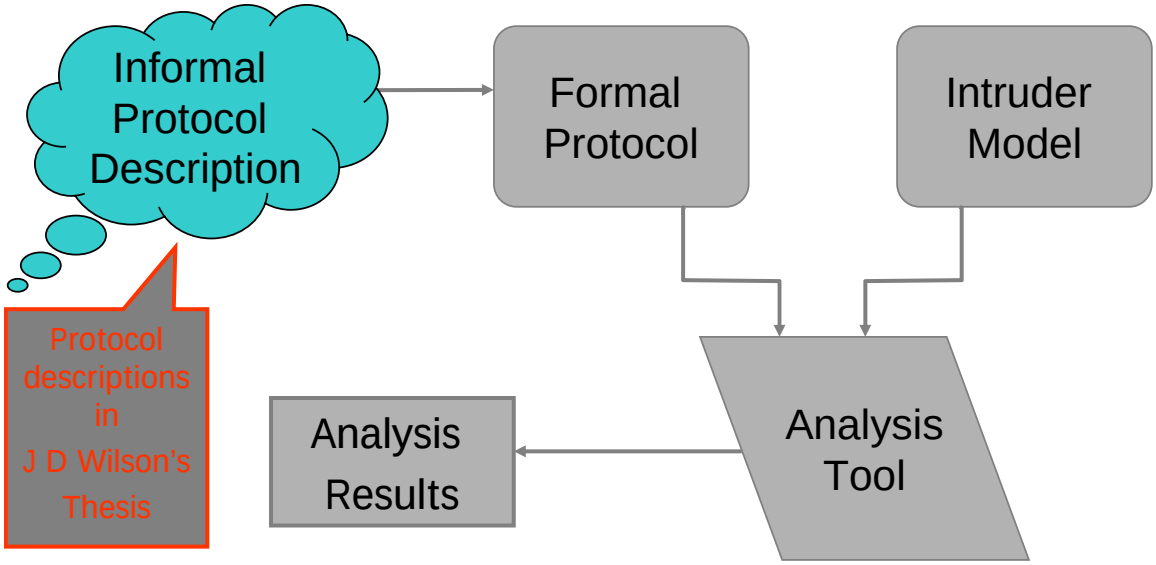


Figure 17. Protocol Analysis Process (Adapted From Ref 37)

The first step is to acquire an informal protocol description that is detailed enough to derive a formal protocol representation. The second step is to derive from the informal protocol description a formal protocol description that can be used in the analysis. The third step in the process is to use the same formal language or method to create an intruder model that correctly reflects the environment and abilities of an intruder. The next step is to apply some formal analysis, either manual manipulation or an automated tool, to the formal definitions from the previous steps. The final step is to present the analysis results, which will either give a counter example or prove that the protocol meets the expectations placed on it given the original assumptions.

This document presents the first step of that process: acquiring an informal protocol description that is detailed enough to derive a formal protocol representation. Acquiring the appropriate request for comments (RFC) is often all that necessary for this step.[37] That is not an option for the present suite of protocols because they are not presented in that format.

A. REFERENCE NUMBERING SYSTEM

The best way to insure that the information presented in JD Wilson's thesis is accurately reflected in the formal representations is to create a mapping. This paper will use the term information unit (IU) to mean the smallest unit that has meaning relevant to the current context. In creating a mapping from one item to another it is important to use the appropriate granularity. The first challenge in creating this mapping is to determine the level of granularity to use for an IU. If the granularity of the IU is too coarse, important details will be lost. If the granularity is too fine, then inconsequential details will overwhelm the effort. So, what constitutes a single IU in JD Wilson's thesis? In JD Wilson's thesis some pieces of information are presented using several sentences while some pieces of information are presented using only one sentence. Therefore, a single English sentence will be considered a single IU for mapping purposes.

The next challenge is to determine how to reference an individual IU. Creating an IU reference numbering system solves this problem. This IU reference numbering system has a few requirements; easy to implement, intuitive, complete, and error resistant. Considering the previous requirements the following number system was developed:

- General Form: IU# = <Page Number>.s<Sentence Number>
- Example: IU# 79.s05
- The above example references the fifth sentence on page 79.

One might wonder why the page number is incorporated into the IU reference number system. The IU reference number system would be simpler if the IU were just numbered sequentially. This solution should be considered because it would allow

someone who has the original thesis in a different format to follow the reference with ease. However, this benefit is outweighed by a single disadvantage: if there is a single error somewhere in the numbering then every number after the error would be incorrect and make use of the number system invalid. Considering that the numbering system is to be implemented by hand the possibility of a simple numbering error is very high. Including a page number makes the system more error tolerant by limiting the affect of an error to a single page. Therefore a numbering system that incorporates page numbers into the reference number system is a better choice. Additionally, the concern about other forms of the thesis is mitigated by the fact that there is only one authoritative form of the thesis that is readily available.

Since the IU reference system is based on sentences and incorporates the page number several other considerations must be addressed to implement this system:

- Sentences that Span Multiple Pages
- Non-Sentence Structures
- Title Pages, Tables of Contents, Blank Pages and Other Document Structures
- Figures and Tables

The original numbering follows the same numbering conventions as footnotes; sentence numbers are written above the period of the sentence. Since the IU reference numbering system is "page based", a sentence is numbered according to its location on the page on which its period is placed. This adds to the simplicity of the system. Every IU number can be found on the page that is contained in the IU number itself. An example is given in Figure 18.

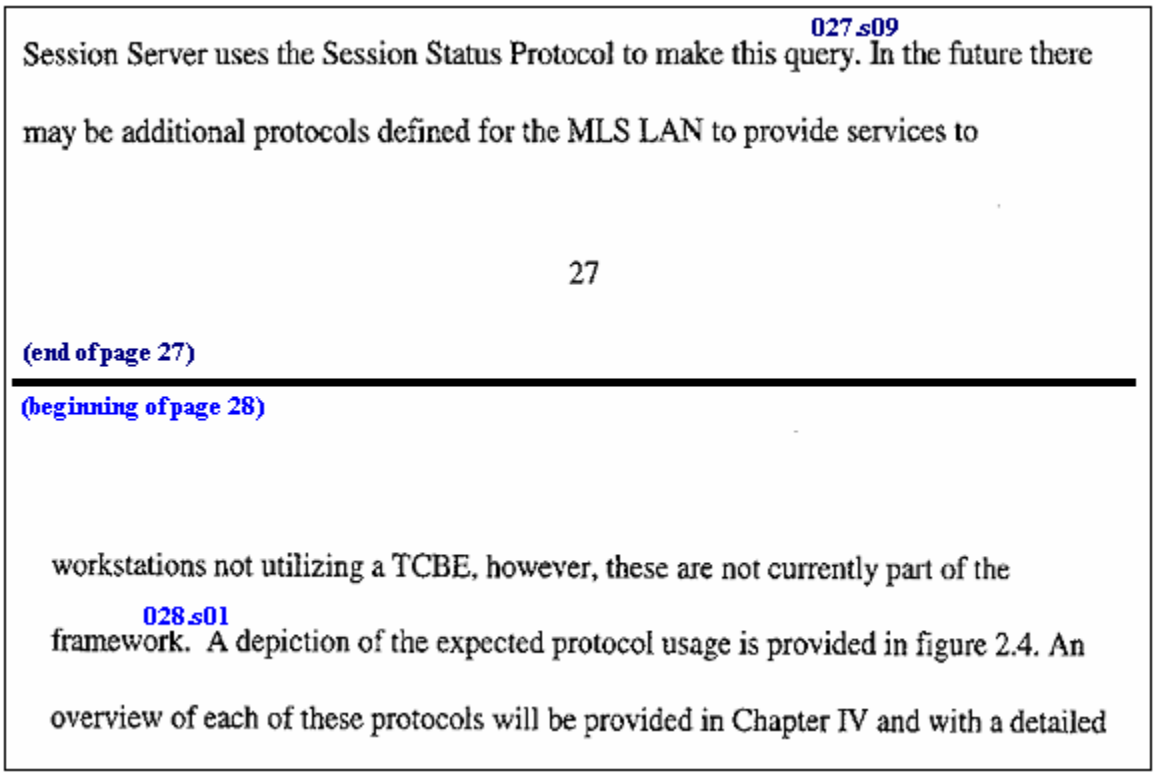


Figure 18. Example of reference numbering across pages.

The sentence that follows the IU reference number 027.s09 starts on page 27 with the words "In the future…" and ends on page 28 with the words "…the framework", therefore it has the IU reference number 028.s01 rather than 027.s10.

There are two non-sentence structures the IU reference number system needs to handle; those that end in a period and those that do not end in a period. Both structures are used in JD Wilson's thesis as titles or other parts of "document structure" but do not in themselves present additional information and therefore they are ignored.

The reference number system needs to be complete and intuitive. A sequential listing of IU numbers should be intuitively complete or incomplete by human inspection. Therefore, entire pages that have no individual IUs, such as title pages, tables of contents, and blank pages are treated as a single IU and given the following format:

- General Form: IU# = <Page Number>.s00
- Example: 50.s00
- The example states that page 50 does not contain information units.

Figures and tables are an important part of the information presented in JD Wilson's thesis. They are given special IU notations because they contain more information that a normal IU. However, because of their unique structure they are treated as a single IU.

- General Form: IU# = <Page Number>.g<graphical number>
- Example: 105.g02
- The above example references the graphical IU which is the second chart/Figure on page 105.

B. INFORMATION UNIT (IU) CLASSIFICATIONS

Each IU in JD Wilson's thesis presents a piece of information. In order to facilitate the analysis of this information each IU is mapped to a label according to the type of information it presented. The impetus for this mapping is to allow the formal process to focus on the IUs that contain information directly related to the policies, requirements, and specifications presented. The IU classifications are as follows:

- Definition – Gives a definition for a term
- Document Structure – Presents information that only deals with the structure of the Document
- Extended IU – Chart or Figure
- Future – Information about future work
- Reiteration – This IU is equivalent to another more authoritative IU
- Requirement – Gives information that pertains to a requirement
- Specification – Gives information that pertains to a specification
- Policy – Gives information that pertains to a policy

C. MAPPING TO CONCISE DESCRIPTIONS

The next step in the process could be considered a mapping or a reduction. Each IU is revisited and mapped to a concise representation of the essential information contained in the unit. Sometimes this information is best conveyed in a sentence

fragment, a list, or even a single word. While other times the original sentence is deemed to be the most concise way to convey the information. There are two reasons for this mapping. The first is to reduce the IU so that the IU reference list will only contain the essential information. English sentences don't always contain just the facts. They sometimes contain additional words, phrases, and ideas to tie the information to previous sentences and the general flow of the document. While theses additions are necessary for good writing, they are not necessary for the purposes of the intended mappings and therefore it is advantages to simplify the IU by removing these additions. The second reason to map to concise descriptions is to facilitate comparisons between IUs. Once IUs are written in a concise manner it becomes easier to determine if two different IUs are equivalent, complementary, or contradictory.

It is very important that the concise descriptions are simply a concentration of the original information that is contained in the original IU. To ensure this requirement, an addition step is taken. Each concise description is mapped to an IU classification. This is done without reference to the original IU classification. The concise description's IU classification is compared to the original IU classification in order to ensure essential information is neither lost nor inserted inadvertently.

D. REDUCTION OF INFORMATION

JD Wilson's thesis "using a realistic Systems Requirements Document and a High Level Protocol Analysis . . . presents a framework of communications protocols".[1] This presentation does a good job of educating the reader about the protocols by repeating the information in various forms and revisiting difficult points. In addition, background information is presented and the document is structured to help reinforce the presentation of the information. While this is an excellent way to present information and educate a general reader, this repetition and supporting material inhibit the creation of a concise mapping. Therefore, the IU reference list collected from JD Wilson's thesis is put through a reduction process. The first step in the process is to remove information that is reiterated in multiple locations. Two important questions need to be addressed before this process can proceed:

- Given two IUs that present the same information, how should this information be "reduced"?
- How should this "reduction" be reflected in the IU reference numbering system?

If two IUs are equivalent then either IU could be used to represent that information. However, in order to implement the reduction in a simple and organized manner the UIs are given an authoritative structure. The collection of IUs presented in JD Wilson's thesis can be broken into four separate categories which correspond to the four documents. Each section has a different purpose and publication date. This allows one to establish an authoritative framework which this reduction process will follow. The following is a list of the sections of the document from most authoritative to least authoritative:

- MLS LAN Systems Requirement Document (pages 81 – 96)
- MLS LAN Protocol High Level Analysis Document (pages 97 – 112)
- MLS LAN Connection Framework Document (pages 113 – 152)
- Thesis Body (pages 1 – 80)

Additionally, if two IUs are in the same authoritative level the lower IU number is more authoritative. Given the authoritative framework described above the reduction process can be expressed in the following guideline: If two IUs are equivalent, the most authoritative IU reference list entry will remain unchanged and the least authoritative will change its IU classification to <Reiteration> and its concise description to the IU number of the more authoritative IU. At the end of this reduction process, the IU reference list will contain a representation of the information presented in JD Wilson's thesis. More importantly for this paper the IU reference list will contain a representation of the information presented in three areas: Policy, Requirements, and Specifications.

E. THE IU LISTING BY IU REFERENCE NUMBER

IU Number	Classification	Description
001.s01	Thesis Goal	Thesis defines a communications framework
001.s02	Thesis Goal	Present network security architecture
001.s03	Thesis Goal	repeat
001.s04	General Info	Justification for MLS System
001.s05	General Info	Justification for MLS System
001.s06	General Info	Justification for MLS System
001.s07	General Info	Justification for MLS System
001.s08	General Info	Justification for MLS System
001.s09	General Info	Justification for MLS System
001.s10	General Info	Justification for MLS System
001.s11	General Info	Justification for MLS System
002.s01	General Info	Justification for MLS System
002.s02	General Info	Justification for MLS System
002.s03	General Info	Justification for MLS System
002.s04	General Info	Justification for MLS System
002.s05	General Info	Justification for MLS System
002.s06	General Info	Four security models
002.s07	General Info	Dedicated security model
002.s08	General Info	Dedicated security model
002.s09	General Info	Dedicated security model
002.s10	General Info	Dedicated security model
002.s11	General Info	Dedicated security model
002.s12	General Info	Dedicated security model
002.s13	General Info	Dedicated security model
002.s14	General Info	System High security model
002.s15	General Info	System High security model
003.s01	General Info	System High security model
003.s02	General Info	System High security model
003.s03	General Info	System High security model
003.s04	General Info	System High security model
003.s05	General Info	System High security model
003.s06	General Info	Compartmented model
003.s07	General Info	Compartmented model
003.s08	General Info	Compartmented model
003.s09	General Info	Compartmented model
003.s10	General Info	Compartmented model
003.s11	General Info	True Multilevel security model
003.s12	General Info	True Multilevel security model
004.s01	General Info	True Multilevel security model
004.s02	General Info	True Multilevel security model
004.s03	General Info	NPS project
004.s04	General Info	high assurance servers
004.s05	General Info	"diskless" personal computers
004.s06	General Info	Network access controlled by TCB
004.s07	General Info	reasonably priced MLS system
004.s08	General Info	Accredible
004.s09	General Info	Criteria Documents
005.s01	General Info	NPS project is "Multilevel Secure"
005.s02	General Info	Multilevel secure network project (MLS LAN)
005.s03	General Info	Multilevel secure networking requirements
005.s04	General Info	Organizational access policy

IU Number	Classification	Description
005.s05	General Info	Three areas
005.s06	General Info	Establish security policy objectives
005.s07	General Info	Set "laws, rules, practices" of policy
005.s08	General Info	"laws, rules, practices" called "Organizational Security Policy"
005.s09	General Info	Automated Security Policy
005.s10	General Info	MLS LAN has some principle guarantees
005.s11	Requirement	MLS LAN absolute control over mechanism that provides data to users
005.s12	General Info	expand 05.s11
006.s01	ISSUE	protection mechanism code, security related processes directly tied to security policy.
006.s02	ISSUE	formal and informal models enforce 06.s1
006.s03	ISSUE	finished processes - shown free of un-validated code
006.s04	Requirement	Trusted code will be validated for necessity
006.s05	Requirement	Trusted code will be validated for sufficiency
006.s06	Requirement	Verifiably ensure - identity and coinciding security of network users
006.s07	Requirement	Verify services to user
006.s08	General Info	benefits of user and information identification
006.s09	Quote	concise restatement of policies in 05.s6-Nov.s8
006.s10	General Info	TCSEC metric info
006.s11	General Info	TCSEC metric info
007.s01	General Info	TCSEC Division D
007.s02	General Info	TCSEC Division D
007.s03	General Info	TCSEC Division C
007.s04	DEFINITION	TCB - pertains to security policy enforcement
007.s05	DEFINITION	TCB - encompasses all security-relevant aspects of the system (network)
007.s06	DEFINITION	DAC - discretionary access control
007.s07	General Info	TCSEC Division B
007.s08	DEFINITION	MAC - Mandatory Access Control
007.s09	General Info	MAC , sensitivity labels
007.s10	General Info	MAC, disclosure parameters
007.s11	General Info	Division B requires a clearly defined security policy model.
007.s12	ISSUE	Security policy model can be either formal or informal
008.s01	General Info	Differences between Divisions
008.s02	Requirement	MLS LAN must satisfy the "Reference Monitor" concept
008.s03	DEFINITION	Reference Monitor
008.s04	General Info	Expand 08.s3
008.s05	General Info	Expand 08.s3
008.s06	General Info	Expand 08.s3
008.s07	General Info	Expand 08.s3
009.s01	Requirement	MLS LAN requires a "trusted path"
009.s02	General Info	Expand 09.s1
009.s03	General Info	Expand 09.s1
009.s04	General Info	Expand 09.s1
009.s05	General Info	TCSEC Division A
009.s06	General Info	TCSEC Division A
009.s07	ISSUE	MLS LAN will be Class B3
009.s08	General Info	TCSEC
009.s09	General Info	TCSEC

IU Number	Classification	Description
009.s10	Thesis Goal	"This thesis will use each of these documents as the basis for its descriptive overview of the MLS LAN's system security, assurance, communications integrity and transmission security features."
010.s01	General Info	MLS LAN non-technical goals
010.s02	General Info	MLS LAN non-technical goals
010.s03	Specification	Previously evaluated Class B3 server
010.s04	Specification	Use TCP/IP
010.s05	General Info	MLS LAN non-technical goals
010.s06	Requirement	Maintain absolute control over data, info and services
010.s07	Requirement	Verifiable protection against disclosure and modification during transmission
010.s08	General Info	
010.s09	General Info	Describe design requirements of components
011.s01	General Info	Describe design requirements of communication between components
011.s02	Thesis Goal	High level overview of components, functionality, and requirements
011.s03	Thesis Goal	Establish communications framework
011.s04	Thesis Goal	Connectivity requirements
011.s05	General Info	Thesis content
011.s06	General Info	Thesis content
011.s07	General Info	MLS LAN - three primary components.
011.s08	DEFINITION	TCB - Trusted Computing Base
012.s01	DEFINITION	TCB - partitioned among MLS LAN components
012.s02	DEFINITION	Network Application Protocol Services - provide functionality for access to available software
012.s03	DEFINITION	Network Computer
012.s04	General Info	Document structure
012.s05	Reiteration	010.s07
012.s06	Requirement	Protected Communications Channel enforces 010.s07
012.s07	General Info	Document structure
012.s08	General Info	Document structure
012.s09	DEFINITION	MLS LAN - connection framework, overview of parameters for initiation, security and communications between components.
012.s10	General Info	Document structure
012.s11	DEFINITION	Level of detail that will be presented in Chapter IV
013.s01	General Info	"Chapter V contains the conclusions made for the use of the proposed architecture and connection framework as defined in the thesis"
013.s02	General Info	Document structure
013.s03	General Info	Document structure
013.s04	QUESTION	
013.s05	General Info	Document structure
013.s06	General Info	Document structure
013.s07	General Info	Document structure
014.s00	Document Structure	Blank
015.s01	DEFINITION	MLS LAN - purpose is to design a trusted network system
015.s02	General Info	describe general network system
015.s03	General Info	Document structure
015.s04	DEFINITION	TNI - Trusted Network Interpretation
015.s05	DEFINITION	Component
015.s06	General Info	Background / supporting information

IU Number	Classification	Description
015.s07	General Info	Background / supporting information
015.s08	General Info	Background / supporting information
015.s09	General Info	Background / supporting information
015.s10	General Info	Background / supporting information
015.s11	DEFINITION	Interconnected Accredited AIS View
015.s12	General Info	expand 15.s11
016.s01	General Info	expand 15.s11
016.s02	General Info	expand 15.s11
016.s03	ISSUE	MLS LAN - uses Interconnected Accredited AIS View
016.s04	ISSUE	Require statement of security policy
016.s05	ISSUE	Require formal security policy
016.s06	General Info	Document structure
016.s07	General Info	Document structure
016.s08	POLICY	MLS LAN - enforces Bell and LaPadula Model
016.s09	General Info	describe BLP
016.s10	General Info	describe BLP
017.s01	General Info	describe BLP
017.s02	General Info	describe BLP
017.s03	General Info	describe BLP
017.s04	General Info	describe BLP
017.s05	General Info	describe BLP
017.s06	General Info	describe BLP
017.s07	General Info	describe BLP
017.s08	General Info	describe BLP
017.s09	General Info	describe BLP
017.s10	General Info	describe BLP
017.s11	General Info	describe BLP
017.s12	General Info	describe BLP
017.s13	General Info	describe BLP
018.s01	General Info	describe BLP
018.s02	General Info	describe BLP
018.s03	General Info	describe BLP
018.s04	Requirement	Biba model - integrity, non-contamination
018.s05	General Info	Describe Biba
018.s06	General Info	Describe Biba
018.s07	General Info	Describe Biba
018.s08	General Info	Describe Biba
018.s09	General Info	Describe Biba
018.s10	General Info	Describe Biba
019.s01	General Info	Describe Biba
019.s02	General Info	Describe Biba
019.s03	General Info	Describe Biba
019.s04	Requirement	Both BLP (inappropriate disclosure), Biba (integrity) are enforced throughout the network
019.s05	Reiteration	011.s07
019.s06	Reiteration	011.s08
019.s07	Reiteration	012.s02
019.s08	Reiteration	012.s03
019.s09	General Info	Document structure

IU Number	Classification	Description
019.s10	Reiteration	007.s06
020.s01	Reiteration	012.s01
020.s02	Reiteration	012.s01
020.s03	DEFINITION	MLS LAN TCB - components built on XTS-300 systems architecture
020.s04	Specification	Security Kernel has complete control of MLS LAN trusted user-developed code
020.s05	Specification	User-developed code, extends TCB to workstations, create secure session application connections, and protect communications
020.s06	Specification	XTS-300 has a four-ring structure
020.s07	Specification	Security domains are enforced in hardware
021.s01	DEFINITION	Commodity Application System Services - CASS
021.s01	DEFINITION	Trusted System Services (TSS)
021.s01	General Info	names of the four primary software components
021.s02	Specification	Security Kernel - Ring 0
021.s03	Specification	Security Kernel - handles Reference Validation Mechanism, MAC, DAC, resource management, process handling, interrupt handling
021.s04	Specification	Ring 1 - (TSS) Controlled by the Security Kernel
021.s04	Specification	Ring 1 - provides networking, I/O, file system management, file system discretionary access policy enforcement for both trusted and untrusted processes
021.s05	Specification	Ring 2 - Trusted software and CASS
021.s06	General Info	administrator security related tasks done at this level (ring 2)
021.s07	Specification	Ring 3 - Untrusted Applications (user)
021.s08	General Info	Document structure
021.s09	DEFINITION	Secure Attention Key (SAK)
021.s09	Specification	XTS-300 - supports SAK recognition and processing, user access identification and authentication, session control and TCP/IP configuration and management
021.s10	Specification	MLS LAN - ring 2 processes: provide extension of the TCB to the TCBE, and the provision of communications protection
022.s01	Specification	MLS LAN TCB - subcomponents are: Extension of TCB to TCBE (ring 2), provision of communications protection (ring 2), protocols defined for connecting two MLS LAN components
022.s02	General Info	Document structure
022.s03	DEFINITION	Protected Communications Channel (PCC)
022.s03	Specification	Protected Channel Initiator - creates the protected communications channel
022.s04	Specification	Protected Channel Initiator - will enforce a "two-way" mutual hardware authentication between the two connecting entities and provide security and integrity protection on all transmitted data
022.s05	Specification	PCC - all other connection protocols operate "within" this conduit
022.s05	Specification	PCC - basis for extending the TCB to distributed components
022.s06	General Info	one logical TCB
022.s07	Specification	Protected Communications Channel (PCC) - provides fault tolerance, network component failure doesn't affect network
022.s08	Future	Protected Channel Initiator
023.s01	General Info	Document structure
023.s02	Specification	Session Database Server - trusted process, manages session status data for each user logged into the MLS LAN

IU Number	Classification	Description
023.s03	Specification	TCB Extension Server - is the only component that can create session status modification requests
023.s04	Specification	Session Status Protocol (SSP) - used for session status modification requests
023.s05	Specification	Session Status Protocol (SSP) - all components may query the information, but the is no write or modification access allowed
023.s06	Specification	Session Status Protocol (SSP) - query, receive current session information on a user
023.s07	Specification	Session Database Server - on XTS-300
023.s08	Future	Loss of communication between TCB Extension Server and Session Database Server could allow unwarranted access to the MLS LAN
023.s09	Future	MLS LAN - requires control mechanism to prevent new connection to the MLS LAN and its services in this even.
023.s10	Future	23.s08-09 are left as future work
023.s11	General Info	Document structure
023.s12	Previous Work	TCB Extension Server process
023.s13	Specification	TCB Extension Server process - extends the TCB perimeter securely over the network to the requesting TCBE-equipped workstation
023.s14	Specification	TCB Extension Server process - only initiated by "secure attention" from user
023.s15	Specification	TCB Extension Server process - single parent and multiple child processes, accepts connections from TCBE-equipped workstations
023.s16	Specification	TCB Extension Server process - parent process listens on assigned port for incoming requests for secure attention
023.s17	Specification	TCB Extension Server process - parent process will verify the identification and authentication of the requesting TCBE
024.s01	Specification	TCB Extension Server process - parent process successful at verification, then child process is forked and given control of the communications
024.s02	Specification	TCB Extension Server process - continues to listen for new connections after child is "handed" the communication
024.s03	Specification	TCB Extension Server process - terminates connections that it can verify
024.s04	ISSUE	Each TCBE connection is assigned an individual child TCB Extension Server process that handles all of the security related operations necessary to establish and maintain a session on the MLS LAN
024.s05	Specification	TCB Extension Server process - child handles trusted path security-related operations
024.s06	Specification	TCB Extension Server process - child controls TCBE with TCBE state commands
024.s07	General Info	Document structure
024.s08	Specification	SAK - can be activated at any time by the user
024.s08	Specification	TCB Extension Server - receives SAK, interrupts current process, verifies the TCBE, begin user login or session negotiation process.
024.s09	General Info	Document structure
024.s10	General Info	Trusted Path considerations
024.s11	General Info	Trusted Path considerations
024.s12	General Info	Trusted Path considerations

IU Number	Classification	Description
024.s13	Requirement	TCB Extension Server - required to update the TCB on all connections and sessions established on the LAN
025.s01	Requirement	TCB - must ensure information used by the TCB entities to establish connections is current and correct
025.s02	Reiteration	SDS -maintains information, TCB Extension Server controls modification of SDS
025.s03	Requirement	TCB Extension Server - modifies the SDS upon a session change
025.s03	Requirement	TCB Extension Server - modifies the SDS upon a TCBE disconnect from the LAN
025.s03	Requirement	TCB Extension Server - modifies the SDS upon a user logout
025.s03	Requirement	TCB Extension Server - modifies the SDS upon initialization of a user session
025.s04	Requirement	Session Database - must be a current depiction of the MLS LAN
025.s05	General Info	Extension Server - TCBE : Trusted path question
025.s06	General Info	during normal LAN operations there is no need for a trusted path
025.s07	General Info	user is operating a previously negotiated level
025.s08	Requirement	SDS - normal operation don t affect the SDS
025.s09	Specification	Application protocol requests query the SDS
025.s10	Requirement	Application Protocol Requests are validated against the TCB's trusted session information
025.s11	Requirement	Application Protocol Requests - that are not commensurate with the user's current session will be denied.
025.s12	Requirement	Session level modifications are done via Extension Server - TCBE trusted path. (SAK initiated)
025.s13	Future	MLS LAN - TCB maintain control over the user's LAN connection
026.s01	Future	TCB - confirm that the user is actually still physically at the terminal
026.s02	Future	26.s1
026.s03	Future	26.s1
026.s04	Future	26.s1
026.s05	General Info	Document structure
026.s06	DEFINITION	TCBE - enhanced network interface card to support a trusted path interface to the user
026.s07	Reiteration	085.s03
026.s08	Specification	TCBE - provides verifiable way to extend the TCB
026.s09	Specification	TCBE - provides SAK mechanism for trusted path initiation and establishes the PCC
026.s10	Specification	TCB-TCBE Connection Protocol - TCB Extension Server - through state commands controls the disk operating system and applications used on the workstation
026.s11	Specification	TCBE - ensures appropriate object reuse between session security levels
026.s12	General Info	Document structure
026.s13	Specification	TCB - uses the defined protocols to establish a session and conduct operations on the MLS LAN
026.s14	Reiteration	022.s03
026.s15	Reiteration	022.s05
027.s01	Specification	After the PCC is established the TCBE must "connect" to the TCB Extension Server for login and session negotiation

IU Number	Classification	Description
027.s02	Specification	TCB-TCBE Connection Protocol - is used for login and session negotiation
027.s03	Specification	TCB Extension Server - updated the SDS user information during session negotiation to reflect the current session
027.s04	Specification	Session Status Protocol (SSP) - supports TCB Extension Servers updating of the SSD through the SDS
027.s05	Requirement	Application Protocol Server - is only accessed after successful session negotiation
027.s06	Specification	Application Protocol Server - is accessed through the Secure Session Server
027.s07	Specification	Session Server Protocol (SSP) - supports requests from application protocol services
027.s08	Specification	Secure Session Server (SSS) - requests SDS to verify users current session information prior to fulfilling user's application protocol request
027.s09	Specification	Secure Session Server (SSS) - uses Session Status Protocol to query SDS
028.s01	Future	Protocols for workstations not using the TCBE are not yet defined
028.s02	General Info	Document structure
028.s03	Specification	MLS LAN - supports multiple simultaneous accesses to higher layer protocol services
028.s04	Specification	TCB - controls access to higher level protocol services in accordance with the security policy
028.s05	Specification	Secure Session Server (SSS) - validates and creates "the connection"
028.s06	DEFINITION	Application Protocol Server (APS) untrusted application layer process that provides a service
028.s07	DEFINITION	Network Application Protocol Services - contains the Secure Session Server and the Application Protocol Server
029.s01	General Info	Document structure
029.s02	Specification	Secure Session Server (SSS) -process - single parent and multiple child processes for each platform on which a given application protocol is based
029.s03	Specification	Secure Session Server (SSS) - processes - are controlled by the security kernel and reside in the Trusted software area
029.s04	Specification	Secure Session Server (SSS) - parent process is responsible for accepting connections from TCBE and establishing TCP/IP service
029.s05	Specification	Secure Session Server (SSS) - parent process listens for requests
029.s06	Specification	Secure Session Server (SSS) - parent process verifies the MLS LAN session with the SDS
029.s07	Specification	Secure Session Server (SSS) - successful verification allows parent to "hand" the child process the communication
029.s08	Reiteration	029.s05
029.s09	Specification	Secure Session Server (SSS) - failure of verification by parent process terminates the session (no child forked)
029.s10	Specification	Secure Session Server (SSS) - each protocol service request is assigned individual child Secure Session Server process which handles all protocol transmissions to and from the APS
029.s11	DEFINITION	Application Protocol Server (APS)
029.s11	Specification	Secure Session Server (SSS) - child process creates unique APS process

IU Number	Classification	Description
029.s12	General Info	Document structure
029.s13	General Info	Document structure
029.s14	Specification	Application Protocol Server process - implements server portion of application level protocol
029.s15	Specification	Application Protocol Server process - support only a single protocol
029.s15	Specification	Application Protocol Server process - untrusted
030.s01	General Info	standard protocol code (with slight functional modification)
030.s02	Specification	Workstations can only communicate with APS through the Secure Session Server (constrained by the underlying TCB)
030.s03	Specification	Workstations - are diskless
030.s04	Specification	TCB - controls the workstations
030.s05	Requirement	Workstations - one logged in user at a time
030.s06	Requirement	Workstations - support up to date OS
030.s07	Future	MLS LAN - supports non-TCBE workstations
030.s08	Future	MLS LAN - allows "anonymous" access to selected application services
031.s01	Specification	MLS LAN - provide protection against disclosure and modification of information on all communications channels used by the network
031.s02	Specification	MLS LAN - uses digital communications encryption
031.s03	General Info	Link vs. End-to-End encryption
031.s04	General Info	link encryption
031.s05	General Info	link encryption
031.s06	General Info	link encryption
031.s07	General Info	link encryption
031.s08	General Info	link encryption
031.s09	General Info	link encryption
031.s10	General Info	link encryption
031.s11	General Info	End-to-End Application-Level Security
031.s12	General Info	End-to-End Application-Level Security
031.s13	General Info	End-to-End Application-Level Security
032.s01	General Info	End-to-End Application-Level Security
032.s02	General Info	End-to-End Application-Level Security
032.s03	General Info	End-to-End Application-Level Security
032.s04	General Info	Transport layer vs. Network layer
032.s05	DEFINITION	Internet Engineering Task Force (IETF)
032.s06	General Info	Document structure
032.s07	DEFINITION	Transport Layer Security (TLS)
032.s07	General Info	Transport layer
032.s08	General Info	Transport layer
032.s09	General Info	Transport layer
032.s10	General Info	Transport layer
033.s01	General Info	Transport layer
033.s02	General Info	Transport layer
033.s03	General Info	Transport layer
033.s04	General Info	Transport layer
034.s01	General Info	Transport layer
034.s02	General Info	Document structure
034.s03	General Info	Transport layer
034.s04	General Info	Transport layer
034.s05	General Info	Transport layer

IU Number	Classification	Description
034.s06	General Info	Transport layer
034.s07	General Info	Transport layer
034.s08	General Info	Transport layer
034.s09	General Info	Transport layer
035.s01	General Info	Transport layer
035.s02	General Info	Transport layer
035.s03	General Info	Transport layer
035.s04	General Info	Transport layer
035.s05	General Info	Transport layer
035.s06	General Info	Document structure
035.s07	General Info	Transport layer
035.s08	General Info	Transport layer
035.s09	General Info	Transport layer
036.s01	General Info	Transport layer
036.s02	General Info	Transport layer
036.s03	General Info	Transport layer
036.s04	General Info	Transport layer
036.s05	General Info	Transport layer
036.s06	General Info	Transport layer
036.s07	General Info	Transport layer
036.s08	General Info	Transport layer
036.s09	General Info	Transport layer
036.s10	General Info	Transport layer
036.s11	General Info	Transport layer
036.s12	General Info	Transport layer
036.s13	General Info	Transport layer
036.s14	General Info	Transport layer
036.s15	General Info	Transport layer
036.s16	General Info	Transport layer
037.s01	General Info	Transport layer
037.s02	General Info	Transport layer
037.s03	General Info	Transport layer
037.s04	General Info	Transport layer
037.s05	General Info	Transport layer
037.s06	General Info	Transport layer
038.s01	General Info	Transport layer
038.s02	General Info	Transport layer
038.s03	General Info	Transport layer
038.s04	General Info	Transport layer
038.s05	General Info	Transport layer
038.s06	General Info	IPSec
038.s07	General Info	IPSec
038.s08	General Info	IPSec
038.s09	General Info	IPSec
038.s10	General Info	IPSec
038.s11	General Info	IPSec
038.s12	General Info	IPSec
038.s13	General Info	IPSec
039.s01	General Info	IPSec

IU Number	Classification	Description
039.s02	General Info	IPSec
039.s03	General Info	IPSec
039.s04	General Info	IPSec
039.s05	General Info	IPSec
039.s06	General Info	IPSec
039.s07	General Info	IPSec
039.s08	General Info	IPSec
039.s09	General Info	IPSec
040.s01	General Info	IPSec
040.s02	General Info	IPSec
040.s03	General Info	IPSec
040.s04	General Info	IPSec
040.s05	General Info	IPSec
040.s06	General Info	IPSec
040.s07	General Info	IPSec
040.s08	General Info	IPSec
040.s09	General Info	IPSec
040.s10	General Info	IPSec
040.s11	General Info	IPSec
041.s01	General Info	IPSec
041.s02	General Info	IPSec
041.s03	General Info	IPSec
041.s04	General Info	IPSec
041.s05	General Info	IPSec
041.s06	General Info	IPSec
042.g01	General Info	IPSec - Implementation Archecture Figure
042.s01	General Info	IPSec
042.s02	General Info	IPSec
042.s03	General Info	IPSec
042.s04	General Info	IPSec
042.s05	General Info	IPSec
042.s06	General Info	IPSec
042.s07	General Info	IPSec
043.g01	General Info	IPSec
043.s01	General Info	IPSec
043.s02	General Info	Document structure
043.s03	General Info	IPSec
043.s04	General Info	IPSec
043.s05	General Info	IPSec
043.s06	General Info	IPSec
043.s07	General Info	IPSec
043.s08	General Info	IPSec
044.s01	General Info	IPSec
044.s02	General Info	IPSec
044.s03	General Info	IPSec
044.s04	General Info	IPSec
044.s05	General Info	IPSec
044.s06	General Info	IPSec
044.s07	General Info	IPSec

IU Number	Classification	Description
044.s08	General Info	IPSec
044.s09	General Info	IPSec
044.s10	General Info	IPSec
044.s11	General Info	IPSec
044.s12	General Info	IPSec
044.s13	General Info	IPSec
044.s14	General Info	IPSec
044.s15	General Info	IPSec
044.s16	General Info	IPSec
045.g01	General Info	IPSec - ESP Packet in Transport Mode Figure
045.s01	General Info	IPSec - IKE
045.s02	General Info	IPSec - IKE
045.s03	General Info	IPSec - IKE
045.s04	General Info	IPSec - IKE
045.s05	General Info	IPSec - IKE
046.s01	General Info	IPSec - IKE
046.s02	General Info	IPSec - IKE
046.s03	General Info	IPSec - IKE
046.s04	General Info	IPSec - IKE
046.s05	General Info	IPSec - IKE
046.s06	General Info	IPSec - IKE
046.s07	General Info	IPSec - IKE
046.s08	General Info	IPSec - IKE
046.s09	General Info	IPSec - IKE
046.s10	General Info	IPSec - IKE
046.s11	General Info	IPSec - IKE
046.s12	General Info	IPSec - IKE
046.s13	General Info	IPSec - IKE
046.s14	General Info	IPSec - IKE
047.s01	General Info	Transport layer vs. Network layer
047.s02	General Info	Transport layer vs. Network layer
047.s03	Requirement	MLS LAN - high assurance network which offers interoperability with COTS application software
047.s04	General Info	Transport layer
047.s05	General Info	IPSec
047.s06	General Info	IPSec
047.s07	ISSUE	MLS LAN - session level information provided to a higher layer application protocol is advisory in nature
047.s08	Requirement	MLS LAN - application protocols are not allowed to enforce security policy
047.s09	Requirement	MLS LAN - each connection to the TCB must have encryption protection that supports sensitivity levels equivalent to or higher than that of the session sensitivity level at which the user is operating
047.s10	Requirement	MLS LAN - different encryption may be used depending on the purpose of the connection
047.s11	Requirement	MLS LAN - connection to the TCB Extension Server for session establishment or renegotiation must be secured sufficiently to support the system high.
048.s01	General Info	Transport Layer Security (TLS)

IU Number	Classification	Description
048.s02	General Info	Multilevel Systems
048.s03	General Info	IPSec - Security Policy Database and Security Association Database
048.s04	Specification	MLS LAN - uses IPSec to define unique security tunnels to specific source hosts
048.s05	Specification	IPSec - initial Security Policy Database will be in non-volatile memory, established by the Security Manager (only allow connections to the TCB Extension Server, disallow all others)
048.s06	Specification	MLS LAN - session must be established, the TCB extension Server can update the TCBE Security Policy Database (SPD) with the security information commensurate with the sensitivity level negotiated on the MLS LAN
048.s07	Requirement	TCBE - will correctly negotiate all other (besides the initial TCB connection) connections to the MLS LAN hosts utilizing the standard Security Association setup of ISAKMP
048.s08	Future	IPSec - remote management of the security policy of IPSec is not covered in the [RFC 2408] however, a trusted agent developed in the TCB could easily create and pass this information through the TCB-TCBE Protected Communications Channel used to negotiate
048.s09	DEFINITION	Domain of Interpretation (DOI)
048.s10	General Info	Security Association (SA) - contains semantics such as "situational identity", "situational secrecy" and "situational integrity"
049.s01	General Info	Security Association (SA)
049.s02	Future	Develop a MLS DOI, based on ISAKMP DOI
049.s02	General Info	ISAKMP DOI does not specifically address multilevel security.
049.s03	Future	Suggestions for the development of the MLS DOI
049.s04	Specification	MLS LAN - uses Network layer security, specifically IPSec
049.s05	General Info	IPSec
049.s06	ISSUE	PCC - this protocol will "secure" separate protocol services between end systems
049.s07	Requirement	Trusted Path can be verifiably secured between the TCB and a TCBE
050.s00	Document Structure	Blank
051.s01	General Info	Document structure
051.s02	General Info	Document structure
051.s03	Specification	PCC - is a Security conduit between two MLS LAN TCB entities
051.s04	Specification	PCC - all MLS LAN protocols must use the PCC to secure their traffic
051.s05	Specification	PCC - uses IP layer security as defined in the IP security Standard for the Internet [RFC 2401]
051.s06	Specification	PCC - enforces "two-way" mutual hardware authentication
051.s06	Specification	PCC - provides "security" and integrity on all transmitted data
051.s07	Reiteration	022.s07
051.s08	Reiteration	022.s07
051.s09	General Info	IPSec - framework is used for the PCC, however the thesis does not attempt to describe its architecture of mechanisms
051.s10	General Info	See RFCs for information on IPSec
051.s11	DEFINITION	Protected Channel Initiator (PCI)
051.s11	Future	Data structures necessary for IPSec implementation (the PCC) have not be finalized
051.s11	Future	Protected Channel Initiator (PCI) - has not been completed

IU Number	Classification	Description
052.s01	General Info	Since 51.s11, the IPSec application is only an "approach" to be taken in the MLS LAN to create a PCC
052.s02	General Info	Document structure
052.s03	General Info	Background / supporting information
052.s04	DEFINITION	Bump-in-the-Stack (BITS) - an IPSec implementation
052.s04	Specification	PCC - uses "Bump-in-the-Stack" IPSec - underneath an existing implementation of the IP protocol stack between the native IP and the local network drivers
052.s05	Specification	PCC - uses "Bump-in-the-Stack" IPSec - which does not require access to the IP source code utilized in the host
052.s06	General Info	Background / supporting information
052.s07	Reiteration	084.s11
052.s07	Reiteration	085.s03
052.s08	Specification	PCC - "user defined trusted code" to be controlled by the Security Kernel
052.s09	POLICY	MLS LAN TCB - each connection to the MLS LAN TCB must be protected in a manner commensurate with the sensitivity of the information transmitted
053.s01	DEFINITION	Security Manager - the person responsible for information assurance at a given site installation of a MLS LAN
053.s01	Reiteration	120.s02
053.s02	Specification	TCB - maintains a table that maps: encryption transform to sensitivity levels that is can support
053.s03	Reiteration	120.s04
053.s04	Reiteration	120.s05
053.s05	DEFINITION	Security Association Database (SAD)
053.s05	Reiteration	120.s06
053.s06	Specification	Security Manager - creates a listing of the specific security parameters that a PCC must enforce for connection to each of the MLS LAN entities
053.s07	ISSUE	TCB - maintains "listing of specific security parameters" that are mapped to potential client session levels
053.s08	Specification	TCB Extension Server knows the SPD assignments for each session level, because the TCB maintains a mapping between [specific security parameters that PCC must enforce for connection] to [potential client session levels].
053.s09	QUESTION	Security Policy Database (SPD) - Initial SPD of the TCBE
053.s09	Specification	Security Manager - establishes the Initial SPD of the TCBE, in non-volatile memory
053.s09	Specification	Security Policy Database (SPD) - Initial SPD of the TCBE will be placed in non-volatile memory
053.s09	Specification	Security Policy Database (SPD) - Initial SPD of the TCBE will only allow the TCBE to apply security and connect to the TCB Extension Server (all other connections are disallowed)
053.s10	Specification	TCB Extension Server - once a session has been established, will update the TCBE SPD with the security connection information commensurate with the sensitivity level negotiated for the session

IU Number	Classification	Description
053.s11	Specification	TCBE - will correctly negotiate all other connections to the MLS LAN utilizing the standard Security Association setup of ISAKMP (Reiteration?)
053.s12	Specification	MLS LAN - Additional encryption algorithms or transforms can be used on the MLS LAN
054.s01	ISSUE	This remote management of the security policy of IPSec is available only because the MLS LAN TCBE can create the initial Protected Communications Channel at system high through the non-volatile Security Policy Database placed on the TCBE
054.s01	Specification	TCBE - non-volatile Security Policy Database placed on the TCBE
054.s02	Future	Non-MLS LAN workstation
054.s03	Future	More information about 54.s2
054.s04	Future	More information about 54.s2
054.s05	DEFINITION	Internet Key Exchange (IKE)
054.s05	Specification	MLS LAN - will use standard IKE to define a key exchange and to negotiate security services to be provided for each PCC
054.s06	General Info	IKE - uses a predefined DOI to outline the required and optional attributes that are negotiated during the phase two exchanges
054.s07	General Info	DOI - is written specifically for use with ISAKMP
054.s08	Reiteration	121.s04
054.s09	Specification	PCC - first PCC established must be between TCBE and TCB Extension Server
054.s10	Specification	PCC - is initiated by the TCBE
054.s10	Specification	user SAK initiates TCBE PCC to TCB Extension Server
055.s01	Specification	Protected Communications Initiator - on the TCBE will use the initial Security Policy Database setting to establish IKE phase One exchanges and establish a secure and authenticated communications channel between the TCBE and the TCB Extension Server host
055.s02	Specification	Protected Communications Initiator - Once the IKE security association (SA) has been established, the phase two negotiations can then be sent to generate the appropriate incoming and outgoing IPSec Sass
055.s03	Specification	Protected Communications Initiator - This exchange (55.s02) negotiates the specific AH and ESP selectors required for each SA
055.s04	Specification	Protected Communications Initiator - selectors are outlined for the unique SA and each entity records the SA information into its Security Association Database under a unique Security Parameter Index
055.s05	Specification	MLS LAN - user login and session negotiation can only be done after PCC has been established between TCBE and TCB Extension Server
055.s06	Specification	TCB Extension Server - issues "PCC update" after successful session establishment, transfers appropriate session level security policy data to the TCBE for inclusion in its SPD, and make available in the SPD the entries for communicating with other MLS LA
055.s07	ISSUE	After 55.s06, user is "Logged in" (at negotiated session level)
055.s08	Specification	PCC - a separate PCC is created (using the PCI) every time an application protocol service is requested
055.s09	Specification	The TCB-TCBE connection protocol is used to provide the Trusted Computing Base (TCB) with a method to conduct security related operations along a trusted path
056.s01	Specification	TCBE gains secure attention from the TCB

IU Number	Classification	Description
056.s01	Specification	TCBE responds to commands of the TCB
056.s02	Specification	TCB-TCBE Connection Protocol - TCB Extension Server sends TCBE state commands through this protocol to control the actions of the TCBE
056.s03	Specification	TCB-TCBE Connection Protocol - is ONLY initiated by Secure Attention request from the user.
056.s04	Specification	PCC - Spoofing attack is handled by PCC
056.s04	Specification	PPC - TCB-TCBE Connection Protocol assumes Replay attack is handled by PCC
056.s05	Specification	TCB Extension Server commands can change ⌠current state of the TCBE÷
056.s05	Specification	TCB-TCBE Connection Protocol - Secure Attention Key can change the current state of the TCBE
056.s06	DEFINITION	Configuration = current state of the TCBE
056.s07	Document Structure	Document Structure
056.s08	Reiteration	124.s07
056.s09	ISSUE	TCB-TCBE Connection Protocol - TCBE state 1 - the power flag is set to false
056.s09	Specification	TCBE state 1=Power Off
056.s10	Specification	TCBE state 2=Idle, the power flag is set to true
056.s11	Future	TCBE state 3=Unprotected Operations
056.s12	Future	System low login
056.s13	Future	Example of System low login
056.s14	Specification	TCBE state 4=Trusted Processing, there is a connection between the TCBE √ TCB to conduct ⌠trusted path operations÷ such as User Identification and Authentication and session negotiations
056.s15	Specification	TCBE state 5=Trusted Session, TCBE is connected to the TCB in association with a specific negotiated user session level
056.s16	Specification	TCBE State: Transition to State 5 causes memory purge.
056.s16	Specification	TCBE State: Transition to State 5 causes new operating system to load
056.s17	Specification	States 5 allows MLS LAN session operations at the negotiated sensitivity level.
056.s17	Specification	TCBE in state 5 allows MLS LAN session operations at the negotiated sensitivity level.
057.s01	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state - response payload type from the TCBE can change current state of the TCB Extension Server
057.s01	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state - Secure Attention Request can change current state of the TCB Extension Server
057.s02	DEFINITION	Configuration = current state of the TCB Extension Server
057.s03	General Info	Document structure
057.s04	Specification	TCB Extension Server - has 6 allowed states
057.s05	Specification	TCB Extension Server State [1]=Power Off
057.s06	Specification	TCB Extension Server State [2]=Idle , power is on listening for Secure Attention Request
057.s07	Specification	TCB Extension Server State [2]=Idle , No connection to the TCBE and Users are not logged in
057.s08	Specification	TCB Extension Server - has made a connection with the TCBE

IU Number	Classification	Description
057.s09	Specification	TBC Extension Server state 3: User I&A can be conducted.
057.s09	Specification	TCB Extension Server - has been extended to the TCBE
057.s10	Specification	TCB Extension Server State [4]=Logged In
057.s10	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state 4 - If TCB Extension Server is in state 4 then TCB Extension Server HAS validated the user has been Identified and Authenticated
057.s11	Specification	TCB Extension Server - All Session Negotiations are done in state 4
057.s11	Specification	TCB Extension Server - state [4] - Session Negotiations are done in TCB Extension Server state 4
057.s11	Specification	TCB Extension Server - The settings negotiated in state 4 are used to establish a MLS LAN session.
057.s12	Specification	In TCBE state 5, the user running trusted session operations in the MLS LAN established their session negotiations in State 4 OR 6 related to this connection
057.s12	Specification	TCB Extension Server State 5=Running
057.s12	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state 5 - the TCB Extension Server has a user running trusted session operations in the MLS LAN
057.s12	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state 5 - the TCB Extension Server is connected to the TCBE
057.s13	Specification	TCB Extension Server State [6]=Trusted Session Processing
057.s13	Specification	TCB-TCBE Connection Protocol - TCB Extension Server - A Secure Attention Request Packet changes TCB Extension Server current state from 5 to 6.
057.s13	Specification	The change of state in the TCB Extension Server caused by a SAR packet does not affect the status of the user trusted session operations
057.s14	Specification	TCB Extension Server - state [6] - Status of the user session can be changed
057.s15	Specification	TCB-TCBE Connection Protocol - Header formats are fixed, has payload field
057.s15	Specification	There is a Payload field in each packet
057.s16	Specification	There are two header formats
058.s01	Specification	TCB-TCBE Connection Protocol - Payload Datagram - used to send information/requests from TCBE to the TCB Extension Server
058.s02	Specification	TCB-TCBE Connection Protocol - Command Datagram - used to send information to the TCBE
058.s03	Document Structure	Document Structure
058.s04	Specification	TCB-TCBE Connection Protocol - TCBE - Password÷ is send to the TCB in a Payload Datagram
058.s04	Specification	TCB-TCBE Connection Protocol - TCBE - User name is send to the TCB in a Payload Datagram
058.s05	Specification	TBCE - Three Payload Packets
058.s06	Document Structure	Non IU
058.s07	Specification	TCB-TCBE Connection Protocol - Secure Attention Request packet (Payload Packet) √ created and sent every time the Secure Attention Key is used.
058.s08	ISSUE	Initialize a Protected Communications Channel if one doesn't already exist.

IU Number	Classification	Description
058.s08	Requirement	The TCBE has a way to keep track of the Protected Communications channel and it can detect if one is already established.
058.s08	Specification	TCB-TCBE Connection Protocol - Secure Attention Request packet (Payload Packet) √ creation will change TCBE current state to state [3] (TP processing)
058.s09	Specification	TCB-TCBE Connection Protocol - Payload Datagram - Response Packet (Payload Packet) √ Response to Command Datagram Packet
058.s10	Specification	The TCBE must be in state 3 (TP processing) when it receives a Command Packet!!
058.s10	Specification	When the TCBE receives a Command Packet from the TCB Extension Server, the TCBE waits (in TP processing state) for user input.
058.s11	Specification	User input determines the time and response to a Command Datagram Packet.
058.s12	Specification	TCB-TCBE Connection Protocol - Payload Datagram - PCC updated packet (Payload Packet) √ is generated after successful creation of the PCC Security Channel Database from the information provided by the TCB Extension Server
058.s13	Specification	TCB-TCBE Connection Protocol - Command Datagram - used to allow the TCB Extension Server to control the actions of the TCBE
058.s13	Specification	TCB-TCBE Connection Protocol - Command Datagram - used to send information to user through the TCBE
058.s14	Specification	TCB Extension Server - has 3 Response Types used in Command Datagrams sent to the TCBE.
059.s01	Specification	TCB Extension Server State can generate a No Response Packet (Command Datagram)
059.s02	Specification	TCB-TCBE Connection Protocol - Command Datagram - LOGOUT is a No Response Packet
059.s02	Specification	TCB-TCBE Connection Protocol - Command Datagram - NOOP (No Operation Expected) is a No Response Packet
059.s02	Specification	TCB-TCBE Connection Protocol - Command Datagram - RUN is a No Response Packet (Command Datagram)
059.s03	Specification	TCB Extension Server State can generate a Response with Echo Packet (Command Datagram)
059.s03	Specification	TCBE echoes the users response to the screen in response to a Response with Echo Packet (Command Datagram)
059.s04	Specification	TCBE - Uses ECHO to echo user input to the screen
059.s05	Specification	TCB Extension Server - "response without Echo" does not echo user input to the screen
059.s06	Reiteration	059.s05
059.s07	Specification	TCB Extension Server - Command field used to "control the actions of the TCBE" AND pass information to the user
059.s08	Specification	There are seven command typesNOOP - 59.s09LOGOUT - 59.s12RUN - 59.s13RESUME - 60.s02NEW - 60.s06 (Future Work)DISCONNECT - 60.s"12"""
059.s09	DEFINITION	No Operation (NOOP)
059.s10	Specification	TCB-TCBE Connection Protocol - TCB Extension Server - Command Datagram - NOOP command is to provide the user with an interactive login and session negotiation with the TCB
059.s11	Specification	TCBE - Displays the payload of a NOOP command without modification

IU Number	Classification	Description
059.s12	Specification	TCB-TCBE Connection Protocol - Command Datagram - LOGOUT command - TCBE purges existing operating system and files, returns to an idle state
059.s13	Specification	TCB-TCBE Connection Protocol - Command Datagram - RUN command - TCBE transitions to state [4](trusted session) with "sanitized" version of the Operating System
059.s14	Specification	TCB-TCBE Connection Protocol - Command Datagram - RUN command - payload displayed to user
060.s01	Specification	TCB-TCBE Connection Protocol - Command Datagram - RUN command - activate a session with the TCBE
060.s02	Specification	TCB-TCBE Connection Protocol - Command Datagram - RESUME command - used to "reactivate" a session with a TCBE
060.s03	Specification	TCB-TCBE Connection Protocol - Command Datagram - RESUME command - TCBE transitions to State [4] (Trusted Session)
060.s04	Specification	TCB-TCBE Connection Protocol - Command Datagram - RESUME command - payload echoed to user
060.s05	Specification	TCB-TCBE Connection Protocol - Command Datagram - RESUME command - TCBE maintains "original" version of the OS and return to the "user's pervious session configurations"
060.s06	Future	NEW command
060.s07	Future	NEW command
060.s08	Future	NEW command
060.s09	Future	NEW command
060.s10	Future	NEW command
060.s11	Future	NEW command
060.s12	Specification	TCB-TCBE Connection Protocol - Command Datagram - DISCONNECT command - "terminates" connection to the TCB Extension Server, TCBE transitions to state [1] (Idle)
060.s13	Specification	TCB-TCBE Connection Protocol - Command Datagram - DISCONNECT command - payload displayed to user
060.s14	Specification	TCBE will terminate the "connection" to the TCB Extension Server when it receives a DISCONNECT command packet
060.s15	Specification	TCB-TCBE Connection Protocol - Command Datagram - UPDATE PCC - TCBE will modify the TCBE's security database with the data contained in the packet's payload area.
060.s16	Specification	TCB-TCBE Connection Protocol - Command Datagram - UPDATE PCC will only be used with Response with Echo or Response without Echo
060.s17	Reiteration	056.s03
060.s17	Reiteration	056.s03
060.s18	ISSUE	Secure Attention Key - establish a Protected Communications Channel, before anything else.
060.s18	Specification	TCBE can only send SAR packet ONLY to the TCB Extension Server
060.s18	Specification	TCB-TCBE Connection Protocol - Secure Attention Packet is sent after the PCC is established.
060.s19	Specification	TCB Extension Server responds to a SAR packet by sending a "series of" NOOP commands to request (username and password.)
061.s00	Specification	username prompt uses Response with Echo Packet
061.s01	Specification	TCB-TCBE Connection Protocol - Command Datagram - password prompt uses Response without Echo Packet

IU Number	Classification	Description
061.s02	Specification	TCB-TCBE Connection Protocol - TCB Extension Server - If user I&A are successful TCB Extension Server generates RE(NOOP)(User Menu Interface)
061.s03	Reiteration	059.s10
061.s04	Specification	TCB-TCBE Connection Protocol - Command Datagram - Menu selections from the (user Menu Interface) are associated with "trusted processes"
061.s04	Specification	TCB-TCBE Connection Protocol - Command Datagram - Menu sent in the RE(NOOP)(User Menu Interface) contains selections
061.s05	Document Structure	Non IU
061.s06	Specification	TCB-TCBE Connection Protocol - selection of Session from the user menu - provides the user with current session information.
061.s07	Specification	TCB-TCBE Connection Protocol - Change Session Level - provides user interactive negotiation of session level
061.s08	Specification	TCB-TCBE Connection Protocol - Change Group - provides user interactive negotiation of group setting
061.s09	Specification	TCB-TCBE Connection Protocol - Logout - end session with MLS LAN
061.s10	Specification	TCB-TCBE Connection Protocol - Command Datagram - Run - use current session parameters and enter "Trusted Session Operations"
061.s11	Specification	TCB Extension Server - answers PL(Session) packet with RE(NOOP) ("Prompts")
061.s12	Specification	TCBE - receives RE(NOOP)(Level Change Prompt) and waits for user selection
061.s13	Specification	TCB-TCBE Connection Protocol - Change Group - provides user interactive negotiation of group setting
061.s13	Specification	TCB-TCBE Connection Protocol - Change Session Level - provides user interactive negotiation of session level
061.s14	Specification	TCB-TCBE Connection Protocol - Change Group - interactive negotiation does not change TCB Extension Server state
061.s14	Specification	TCB-TCBE Connection Protocol - Change Group - interactive negotiation does not change TCBE state
061.s14	Specification	TCB-TCBE Connection Protocol - Change Session Level - interactive negotiation does not change TCB Extension Server state
061.s14	Specification	TCB-TCBE Connection Protocol - Change Session Level - interactive negotiation does not change TCBE state
061.s15	Specification	TCB-TCBE Connection Protocol - Change Group - Information sent to the user will be sent via RE(NOOP)(<user information>) packets
061.s15	Specification	TCB-TCBE Connection Protocol - Change Session Level - Information sent to the user will be sent via RE(NOOP)(<user information>) packets
061.s16	Specification	PL(<item from the state [3] menu>) packets can only be sent when the TCBE is in state [3] TP processing
061.s16	Specification	TCB-TCBE Connection Protocol - Change Session Level - TCBE will send user input via PL(<user response or input>)
062.s01	Specification	TCB Extension Server - Answers PL(Logout) packet with NR(logout)(<?>) packet
062.s02	Specification	TCB Extension Server - Answers PL(Run) by starting the "process to establish a session on the MLS LAN"
062.s03	Specification	Step one of the "process to establish a session on the MLS LAN" is for the TCB Extension Server to update the TCBE's Security Database

IU Number	Classification	Description
062.s04	Specification	TCB Extension Server sends a RWOE(Update PCC) packet to the TCBE
062.s05	Specification	RWOE(Update PCC)(<database information necessary for the TCBE to negotiate future Protected Communications Channels at the currently negotiated session level>)
062.s06	Specification	When the TCBE completes an update to its Security Policy Database it sends a PCC Updated response packet
062.s07	Specification	TCB Extension Server MUST receive a PL(PCC Updated) packet before it will generate and send a NR(Run)()
062.s08	Specification	TCBE - receives NR(RUN)() packet and purges the present OS, Load OS, enter trusted operations
062.s09	ISSUE	TCB-TCBE Connection Protocol - Secure Attention Key - will suspend any other process, and send a PL(SAR) packet
062.s10	Specification	TCB Extension Server - receives PL(SAR), it will stop current process and enter I&A portion of the MLS LAN login
062.s11	Specification	TCB-TCBE Connection Protocol - TCB Extension Server - IF successful login process* TCB Extension Server generates and sends NR(Disconnect)()
062.s12	Specification	TCB - receives NR(Disconnect)() packet, "terminates connection with TCB"
062.s13	ISSUE	Once conducting the Trusted Operations no change to the TCB configuration without use of the Secure Attention Key
063.s01	Specification	TCB Extension Server - "knows user is logged and running current session. So It gives an additional selection in the User Interface menu.
063.s02	ISSUE	Resume - Allows user to return to his previous negotiated session without change
063.s03	Specification	Session Status Protocol (SSP) - Session Server Database entries contain: user identification, the TCBE the user is using, sensitivity and integrity levels for the current session
063.s04	DEFINITION	Session Database Server (SDS)
063.s05	Specification	Session Status Protocol (SSP) - TCB Extension Server - only TCB entity that has both read and write access to the SDS
063.s06	Specification	Session Status Protocol (SSP) - used by all TCB entities to read SDS entries
063.s07	Reiteration	063.s06
063.s08	Specification	PCC - Session Status Protocol (SSP)- assumes PCC handles Replay and Spoofing
064.s01	Reiteration	064.s12
064.s02	Specification	TCB Extension Server - can write to SDS only from the following states:State[2] (Connected)State[3] (Logged In)"State[5] (Trusted Session Processing)"""
064.s03	Unclear	TCB entities can request SDS info at anytime??
064.s04	Specification	Session Status Protocol (SSP) - SSP Datagrams don't constitute a state transition for any TCB Entity
064.s05	Specification	Session Status Protocol (SSP) - Session Database Server (SDS) - is stateless as far as the Session Status Protocol is concerned
064.s06	Specification	Session Status Protocol (SSP) - Has fixed header formats, followed by a payload field

IU Number	Classification	Description
064.s07	Specification	Session Status Protocol (SSP) - Has two header formats:Request Datagram, Replay Datagram
064.s08	Specification	Session Status Protocol (SSP) - Request Datagram - used to send requests from a TCB entity to the SDS
064.s09	Specification	Session Status Protocol (SSP) - Replay Datagram - used to respond to TCB Entity's request
064.s10	General Info	Document structure
064.s11	Specification	Session Status Protocol (SSP) - Request Datagram - (list) request of the SDS
064.s12	Specification	TCB Extension Server - only entity that has write, create, modify access to the SDS
064.s13	Specification	Request Datagram - has four commands: List - 65.s01, Create - 65.s03, Modify - 65.s05, Delete - 65.s07
065.s01	Specification	Session Status Protocol (SSP) - Request Datagram - List command - SDS returns attribute values contained under "User Session Identification" number
065.s02	Specification	Session Status Protocol (SSP) - Request Datagram - List command - SDS response to this command reflects that user is / is not logged in
065.s03	Specification	Session Status Protocol (SSP) - Request Datagram - Create command - directs SDS to create a new entry in the SDS.
065.s04	Specification	Session Status Protocol (SSP) - Request Datagram - Create command - TCB Extension Server uses payload field value to pass user and session information to the SDS
065.s05	Specification	Session Status Protocol (SSP) - Request Datagram - Modify command - directs SDS to modify an entry in the SDS
065.s06	Specification	Session Status Protocol (SSP) - Request Datagram - Modify command - TCB Extension Server uses payload field value to pass user and session information to the SDS
065.s07	Specification	Session Status Protocol (SSP) - Request Datagram - Delete command - directs SDS to delete a current record in the database
065.s08	Reiteration	064.s09
065.s09	Specification	Response Datagram - has three response types:ACK response, NAK response, Payload response
065.s10	Specification	Session Status Protocol (SSP) - Response Datagram - ACK response - when TCB Entity only needs success notification
065.s11	Specification	Response Datagram - ACK response - response to CREATE, MODIFY, DELETE
065.s12	Specification	Session Status Protocol (SSP) - Response Datagram - ACK response - payload carries "success verification information" for the TCB Extension Server
065.s13	Specification	Session Status Protocol (SSP) - Response Datagram - NAK response - used when TCB Entity requires failure response
065.s14	Specification	Session Status Protocol (SSP) - Response Datagram - NAK response - response to CREATE, MODIFY, DELETE, LIST
065.s15	Specification	Session Status Protocol (SSP) - Response Datagram - NAK response - payload carries "reason for failure"
065.s16	Specification	Session Status Protocol (SSP) - Response Datagram - Payload response - returns record information to requesting TCB Entity"This is used to respond to a Request Datagram - List command"""

IU Number	Classification	Description
065.s17	General Info	65.s16
066.s01	Specification	Session Status Protocol (SSP) - Request Datagram - List command - TCB Entity - receives request for Network Application Services, generates LIST Request Packet placing requestor's TCBE ID in the User Session Identification field
066.s02	Reiteration	065.s01
066.s03	Reiteration	065.s16
066.s04	Reiteration	065.s02
066.s05	Specification	Session Status Protocol (SSP) - Response Datagram - Payload response - returns record information to requesting TCB Entity, "This is used to respond to a Request Datagram - List command - and allows the TCB entity to continue the connection process"""
066.s06	Specification	Session Status Protocol (SSP) - Request Datagram - List command - Response Datagram of NAK terminates the Application Protocol connection
066.s07	Reiteration	064.s12
066.s08	Specification	TCB Extension Server - receives SAR then it transmits a Request Datagram LIST command
066.s09	Specification	Session Status Protocol (SSP) - Request Datagram - List command in 66.s08 is used to determine if TCBE is already in the SDS
066.s10	Reiteration	065.s02
066.s11	Specification	TCB Extension Server - receives Response Datagram - Payload Datagram causes the TCB Extension Server to transition to State[3] (Logged in)
067.s01	Specification	TCB Extension Server - receives Response Datagram - NAK and continues with user I&A, remains in current state
067.s02	Reiteration	065.s03
067.s03	Reiteration	065.s04
067.s04	Specification	TCB Extension Server - must issue a CREATE command before the TCB Extension Server transitions into state[3](Logged In)
067.s05	Specification	Session Database Server (SDS) - receives Request Datagram CREATE (from TCB Extension Server) responds with an Response Datagram - ACK command upon completion
067.s06	Specification	Session Database Server (SDS) - receives Request Datagram CREATE (from TCB Extension Server) responds with an Response Datagram - NAK command upon error
067.s06	Specification	TCB Extension Server - receives NAK from SDS, sends a retransmission of the CREATE
067.s07	ISSUE	If connection to the SDS is lost the TCB Extension Server will initialize a "command mechanism" to prevent all further connections to the MLS LAN or its services
067.s08	Future	TCB Extension Server "command mechanism"
067.s09	Specification	TCB Extension Server - "Logged In" state allows user to negotiate a session through the TCB-TCBE connection protocol
067.s10	Specification	TCB Extension Server - Receives TCB-TCBE RUN, sends MODIFY command to SDS

IU Number	Classification	Description
067.s11	Specification	Session Status Protocol (SSP) - Session Database Server (SDS) - uses command field of MODIFY to modify current database entry (meaning when the SDS recieves a Request packet "MODIFY" it will modify the SSD)
067.s12	Specification	Session Database Server (SDS) - receives Request Datagram MODIFY (from TCB Extension Server) responds with an Response Datagram - ACK command upon completion
067.s13	Specification	Session Database Server (SDS) - receives Request Datagram MODIFY (from TCB Extension Server) responds with an Response Datagram - NAK command upon error
067.s13	Specification	TCB Extension Server - receives NAK from SDS, sends a retransmission of the MODIFY
068.s01	ISSUE	TCB "DISCONNECT" completes a user session
068.s01	Specification	TCB Extension Server - receives TCB "DISCONNECT", sends Request packet "DELETE" command to SDS
068.s01	Specification	TCB Extension Server - receives TCB-TCBE payload packet "LOGOUT", sends Request packet "DELETE" command to SDS
068.s01	Specification	TCB-TCBE Connection Protocol - Payload Datagram containing "LOGOUT" completes a user session
068.s02	Reiteration	065.s07
068.s03	Reiteration	065.s12
068.s04	Reiteration	065.s13
068.s05	Specification	TCB Extension Server - Success of "DELETE" packet doesn't affect the success of the user logout.
068.s06	Requirement	MLS LAN will provide access to multiple Application Protocols
068.s07	Requirement	Application Protocols are only accessible to users who have logged in to the MLS LAN and established a session with the TCB
068.s08	DEFINITION	Secure Session Server (SSS)
068.s08	Specification	TCBE-to-Session Server Connection Protocol - Is provided as a method for the TCBE to pass a unique identifier to the Secure Session Server (SSS) in order for it to check with the Session Database Server (SDS) for the users information, TCBE -> (unique id
068.s09	Specification	MLS LAN uses TCBE Identification Number as a unique identifier
068.s10	Specification	TCBE-to-Session Server Connection Protocol - design should allow alternate future data to be inserted in the unique identifier
068.s11	Requirement	SSS - is responsible for establishing correct session level connectivity to the appropriate MLS LAN Application Protocol Server, based on SDS entries
068.s12	Specification	TCBE-to-Session Server Connection Protocol - SSS - Terminates connections that have no SDS entry
069.s01	Specification	TCBE-to-Session Server Connection Protocol - TCBE - uses this protocol ONLY to pass its unique identifier to the SSS
069.s02	Future	TCBE - using TCBE-to-Session Server Connection Protocol from state[2] (Unprotected Operations)
069.s02	Specification	TCBE-to-Session Server Connection Protocol - TCBE - can only use this protocol from state[4] (Trusted Session)
069.s03	Specification	TCBE-to-Session Server Connection Protocol - TCBE - doesn't transition state using TCBE-to-Session Server Connection Protocol

IU Number	Classification	Description
069.s04	Specification	TCBE-to-Session Server Connection Protocol - SSS - is created for each "higher" layer Application protocol supported by the MLS LAN
069.s05	Specification	TCBE-to-Session Server Connection Protocol - SSS - accepts valid requests for particular protocols
069.s06	Specification	TCBE-to-Session Server Connection Protocol - SSS - TCP/IP Application Protocol connection requests from the TCBE to change the SSS's configuration
069.s07	Specification	TCBE-to-Session Server Connection Protocol - SSS - the configuration of the SSS is not relevant to the use of the TCBE-to-Session Server Connection Protocol
069.s08	Specification	TCBE-to-Session Server Connection Protocol - has single fixed header and variable payload field
069.s09	Specification	TCBE-to-Session Server Connection Protocol - "Identification Datagram" passes TCBE ID to SSS
069.s10	General Info	Document structure
070.s01	Specification	User -> "Application Protocol Service Connection Request" -> TCBE -> "Identification Packet" -> SSS
070.s02	General Info	Additional information about 70.s1
070.s03	Specification	TCBE-to-Session Server Connection Protocol - "Identification Packet" -> SSS -> "List" -> SDS
070.s04	Reiteration	065.s01
070.s05	Specification	TCBE-to-Session Server Connection Protocol - SSS - will continue with the Application Protocol Server Operations only if the List packet is validated
070.s06	Specification	TCBE-to-Session Server Connection Protocol - SSS - will terminate the Application Protocol Server Operations if the user is not logged in
070.s07	ISSUE	What does this affect
070.s08	ISSUE	What does this affect
071.s01	General Info	Background
071.s02	General Info	Background
071.s03	General Info	Background
071.s04	General Info	Background
071.s05	General Info	Background
071.s06	General Info	Background
071.s07	General Info	Background
072.s01	General Info	Background
072.s02	General Info	Background
072.s03	Requirement	MLS LAN - must have the ability to extend the TCB from a high assurance server to a commercial PC
072.s04	General Info	Background
072.s05	General Info	Background
072.s06	General Info	Background
072.s07	General Info	Background
072.s08	General Info	Background
072.s09	General Info	Background
072.s10	General Info	Background
073.s01	General Info	Background
073.s02	General Info	Background
073.s03	General Info	Background

IU Number	Classification	Description
073.s04	General Info	Background
073.s05	General Info	Background
073.s06	General Info	"A recommendation for future engineering team efforts would be to start with the identification of the mission requirements and use these to establish engineering goals
073.s07	General Info	Background
073.s08	General Info	Background
074.s01	General Info	Background
074.s02	General Info	Background
074.s03	General Info	Background
074.s04	General Info	Background
074.s05	General Info	Background
074.s06	General Info	Background
074.s07	General Info	Background
074.s08	General Info	Background
074.s09	General Info	Background
074.s10	Future	Limitation of Session Sensitivity Levels
074.s11	Future	Limitation of Session Sensitivity Levels
074.s12	Future	Limitation of Session Sensitivity Levels
074.s13	Future	Limitation of Session Sensitivity Levels
075.s01	Future	Limitation of Session Sensitivity Levels
075.s02	Future	Acceptance of Non-TCBE-Equipped Workstation
075.s03	Future	Acceptance of Non-TCBE-Equipped Workstation
075.s04	Future	Acceptance of Non-TCBE-Equipped Workstation
075.s05	Future	Acceptance of Non-TCBE-Equipped Workstation
075.s06	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
075.s07	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
075.s08	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
075.s09	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s01	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s02	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s03	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s04	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s05	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
076.s06	Future	Session Domination Algorithm
076.s07	Future	Session Domination Algorithm
076.s08	Future	Session Domination Algorithm
076.s09	Future	Session Domination Algorithm
077.s01	Future	Session Domination Algorithm
077.s02	Future	Protected Channel Initiator

IU Number	Classification	Description
077.s03	Future	Protected Channel Initiator
077.s04	Future	Protected Channel Initiator
077.s05	Future	Distributed Session Database
077.s06	Future	Distributed Session Database
077.s07	Future	Distributed Session Database
077.s08	Future	Distributed Session Database
077.s09	Future	Session Time Control Mechanism
077.s10	Future	Session Time Control Mechanism
078.s01	Future	Session Time Control Mechanism
078.s02	Future	Session Time Control Mechanism
078.s03	Future	Session Time Control Mechanism
078.s04	Future	Session Time Control Mechanism
078.s05	Future	Session Time Control Mechanism
078.s06	Future	Session Time Control Mechanism
078.s07	Future	Session Time Control Mechanism
078.s08	Future	TCB-TCBE Trusted Path Connectivity
078.s09	Future	MLS LAN Domain of Interpretation
078.s10	Future	MLS LAN Domain of Interpretation
078.s11	Requirement	MLS LAN framework is intended to provide protected communications between each of the components of the MLS LAN to ensure single level users can access multilevel data.
079.s01	Reiteration	051.s05 PCC - utilizes IPSec to provide security
079.s01	Requirement	MLS LAN Trusted Path = Protected Communications Channel
079.s02	Specification	TCB-TCBE Connection Protocol - Extends TCB protection and control to TCBE
079.s03	Specification	Session Status Protocol - Allows TCB entities to query the Session Status Database
079.s03	Specification	Session Status Protocol - Allows TCB Extension Server to control the Session Status Database
079.s04	Specification	TCBE-to-Session Server Connection Protocol - Allows TCBE in trusted sessions to access Network Application Protocol Services
079.s05	Requirement	MLS LAN wants to extend TCB to TCBE equipped commercially procured personal computers and securely establish multilevel access across a LAN
079.s06	General Info	Document structure
080.s00	Document Structure	Blank
081.s01	General Info	Document structure
081.s02	General Info	Document structure
081.s03	General Info	Document structure
081.s04	General Info	Document structure
081.s05	General Info	Document structure
082.s00	Document Structure	Title Page
083.s00	Document Structure	Table of Contents
084.s01	General Info	Document structure
084.s02	General Info	Background
084.s03	General Info	Background
084.s04	General Info	Background
084.s05	General Info	Background
084.s06	General Info	Background

IU Number	Classification	Description
084.s07	General Info	Background
084.s08	General Info	Background
084.s09	Requirement	Goal of the MLS LAN Project - cons effective, multilevel, office environment leveraging existing high assurance tech.
084.s10	ISSUE	MLS LAN - POLICY
084.s11	Specification	MLS LAN - Uses XTS-300 Server
084.s12	Requirement	MLS LAN - allow separation of users who are at different clearance levels and prevents lower level user from reading a higher level user's files or data
084.s12	Specification	MLS LAN - XTS-300 provides mandatory and discretionary access controls
084.s13	Specification	XTS-300 - establishes "Trusted computing Base", TCB System Services, and security kernel
084.s14	Specification	Security Kernel - implements the TCSEC defined Reference Monitor
085.s01	DEFINITION	Trusted Computing Base Extension (TCBE)
085.s01	Requirement	MLS LAN - logically isolated and unmistakably distinguishable trusted communication path between the server and its clients through development of a Trusted Computing Base Extension TCBE
085.s02	Requirement	TCBE - provides a trusted network interface entity for verifiable expansion of the TCB to the client workstation
085.s03	Specification	TCBE - uses Intel I960jx
085.s04	Requirement	TCBE - Will dominate all actions of the untrusted workstation and allow connectivity into the High Assurance LAN only following the establishment of a trusted path
085.s05	DEFINITION	MLS LAN user - any user who accesses the MLS LAN
085.s06	DEFINITION	TCB Authenticated user - user that has successfully established a TCB-to-User connection and been validated by the TCB for operations within the MLS LAN
085.s07	Future	Non-TCB Authenticated User
085.s08	Future	Non-TCB Authenticated User
085.s09	Specification	MLS LAN - has three components: TCB, Network Application Protocol Services, Workstation
085.s10	Requirement	TCB - provides a fixed security perimeter for MLS LAN operations
085.s11	Requirement	Network Application Protocol Services - provides network functionality for access to available application software
085.s12	Requirement	Workstation - acts as an agent for the User to access any required network functionality
085.s13	DEFINITION	Trusted Computing Base Extension (TCBE) - is a abstraction for the collection of elements of a computer system that pertain to the security policy
085.s14	DEFINITION	Trusted Computing Base Extension (TCBE) - encompasses all policy enforcement, auditing, identification and authentication, and interface for security administration
085.s15	General Info	Document structure
085.s16	Requirement	TCB - can be securely extended to users
085.s17	DEFINITION	Operating System Services (OSS)

IU Number	Classification	Description
085.s17	Specification	XTS-300 - enables MLS LAN to place a trusted daemon process in the Operating System Services (OSS) Domain which provides the protection and communications protocols necessary to establish a trusted path between the workstation and the MLS LAN
086.s01	Requirement	TCB Extension Server - extends the TCB perimeter securely over the network to the requesting TCBE
086.s02	Requirement	TCB Extension Server - provides user identification and authentication, session negotiation, session activation, and session termination
086.s03	Specification	Secure Session Server (SSS) - is a trusted daemon "server" process in the OSS
086.s04	Requirement	Secure Session Server (SSS) - will only accept Network Application Protocol Service requests from workstations that have established a session via the trusted path and the TCB Extension Server
086.s05	Specification	Validated requests will be passed on to untrusted application Protocol servers, operating on behalf of the user, at the user's negotiated session sensitivity level
086.s06	Future	anonymous user
086.s07	Future	anonymous user
086.s08	Requirement	MLS LAN - trusted database maintains all pertinent information concerning each unique TCB session connection
086.s09	Specification	Session Database Server - provides protection for trusted "read" functionality for all TCB entities
086.s09	Specification	Session Database Server - provides protection for trusted "write" of ONLY the TCB Extension Server
086.s10	DEFINITION	Trusted Computing Base Extension (TCBE) - is a hardware-based computer subsystem that is embedded into the MLS LAN workstation
086.s11	Specification	Trusted Computing Base Extension (TCBE) - verifiable high assurance entity that can be used to extend the TCB
086.s12	Requirement	MLS LAN -Connection Protocols, define the parameters for initiation, security and communications establishment between two or more components of the MLS LAN
086.s13	Specification	MLS LAN - Uses TCP/IP stack to support application Layer Protocol Services
086.s14	DEFINITION	Application Protocol Servers (APS)
086.s14	Specification	Application Protocol Servers (APS) - provide access to the Application Layer Protocol Services
086.s15	Requirement	Application Protocol Servers (APS) - are considered untrusted, external to the TCB
086.s15	Requirement	Secure Session Server (SSS) - controls the access to APS, allowing access to data of multiple sensitivity levels
087.s01	DEFINITION	MLS LAN Workstation - network computer used to access MLS LAN
087.s02	Reiteration	087.s03
087.s03	Requirement	MLS LAN - supports simultaneous high assurance success for unique workstations operating at different sensitivity levels
087.s04	Requirement	MLS LAN - provide access to shared resources and application protocol services for Authenticated users
088.s01	Requirement	MLS LAN - provide high assurance connectivity to application protocols that give access to multiple levels of data in accordance with security policies

IU Number	Classification	Description
088.s02	General Info	Document structure
088.s03	General Info	Document structure
088.s04	General Info	Document structure
088.s05	Requirement	TCB - provides a Secure Attention Key(SAK) mechanism to invoke a trusted path from workstations to which the TCB has been extended
088.s06	Requirement	TCB - establishes trusted path between network users and TCB
088.s07	Specification	Trusted Path - will be used for any specified user operations
088.s07	Specification	Trusted Path - will be used for initial session authentication purposes
088.s08	Requirement	TCB-to-TCBE Connection Protocol Channel - If this connection is lost then network functionality will be lost
088.s09	Requirement	TCB - allows users to change sensitivity level (up to configured maximum for that user)
088.s10	Requirement	TCB - security mechanism is always invoked and non-by-passable
088.s11	Requirement	TCB - provide protection against discloser and modification on all network channels
088.s12	Requirement	TCB - shall control access all devices and networks external to the MLS LAN
088.s13	Future	TCB - limit on session sensitivity-level
089.s01	Requirement	TCBE - support Trusted Path
089.s02	Requirement	TCBE - prevent data retention between session security levels (support proper object reuse)
089.s03	Requirement	TCBE - support hardware purge of memory between session security levels
089.s04	Requirement	TCBE - ability to reset host computer system
089.s05	Requirement	TCBE - Support Secure Attention Key
090.s01	Requirement	TCBE - control information flow into and out of the host computer system
090.s02	Requirement	MLS LAN - provide secure communications channel and mutual authentication between TCB entities
090.s03	DEFINITION	Protected Communications Channel (PCC)
090.s03	Requirement	Protected Communications Channel (PCC) - provides secure communications channel and mutual authentication between TCB entities
090.s04	Specification	MLS LAN - all protocols must use the PCC
090.s05	Requirement	MLS LAN - protocol for communication between the TCBE and the TCB Extension Server
090.s06	Specification	TCB-to-TCBE Connection Protocol - should be called, TCB-to-TCBE Protocol - provides for communication between the TCBE and the TCB Extension Server
090.s07	Requirement	MLS LAN - provide secure transfer of information from the TCB Extension Server to the Session Database Server (initialize or modify user session data)
090.s08	Requirement	MLS LAN - provide protocol for a TCB entity to query the Session Database Server for user session information
090.s09	ISSUE	Session Status Protocol
090.s10	Requirement	MLS LAN - provide protocol to support TCBE connection to MLS LAN Secure Session Server
090.s11	Specification	MLS LAN - TCBE-to-Session Server Protocol - conduit for application protocols

IU Number	Classification	Description
090.s12	Future	TCBE to untrusted Application Server
090.s13	Future	Non TCBE workstation to MLS LAN Application Protocol Server
090.s14	Requirement	MLS LAN - support multiple simultaneous accesses to higher layer application protocols
090.s15	Specification	MLS LAN - Application Protocol Servers provide access to shared network resources for TCB authenticated users
091.s01	Specification	MLS LAN - Application Protocol Servers data is accessed only according to security policy
091.s02	Future	Non-TCB authenticated User access to APS
091.s03	Requirement	MLS LAN - support TCBE equipped personal computers
091.s04	Future	Non TCBE workstation support
091.s05	Specification	MLS LAN - workstations support up-to-date Operating Systems
091.s06	Specification	MLS LAN - workstations, TCBE equipped, "diskless thin-client"
091.s07	Requirement	MLS LAN - only one logged in user per workstation at a time
092.s00	Document Structure	Blank
093.s00	Document Structure	Appendix
094.s00	Document Structure	Appendix
095.s00	Document Structure	References
096.s00	Document Structure	Blank
097.s01	General Info	Document structure
097.s02	Requirement	MLS LAN - provide connection protocols to support the extension of the TCB to the user through the TCBE
097.s03	General Info	Document structure
097.s04	General Info	Document structure
098.s00	Document Structure	Title Page
099.s00	Document Structure	Table of Contents
100.s01	General Info	Document structure
100.s02	General Info	Document structure
100.s03	General Info	Document structure
100.s04	General Info	Document structure
100.s05	Reiteration	097.s02
100.s06	Reiteration	097.s03
100.s07	Reiteration	097.s04
100.s08	Reiteration	084.s09
100.s09	Reiteration	084.s10
100.s10	Requirement	MLS LAN - ensure positive control over communications between MLS LAN entities
100.s11	General Info	Document structure
100.s12	Specification	TCB - (PCC) must provide protection against disclosure and modification on all transmissions between entities of the MLS LAN
100.s13	Specification	TCB - (PCC) non-by-passable protected communications channel, provides mutual authentication and data encryption
101.s01	Specification	TCB - (PCC) is a protected conduit through which all other MLS LAN protocols negotiate connectivity
101.s02	Requirement	TCB - session establishment requires user to authenticate themselves to the TCB
101.s03	Requirement	TCB - all security related operations between user and TCB must be conducted through a trusted path

IU Number	Classification	Description
101.s04	Requirement	TCB - "The TCB shall support a trusted communications path between itself and users for use when a positive TCB-to-user connections required (e.g., login, change subject security level)"
101.s05	Requirement	TCB - "Communications via this trusted path shall be activated exclusively by a user of the TCB and shall be logically isolated and unmistakably distinguishable from other paths"
101.s06	General Info	Common Criteria requirements
101.s07	General Info	Common Criteria requirements
102.s01	General Info	Common Criteria requirements
102.s02	General Info	Common Criteria requirements
102.s03	General Info	Common Criteria requirements
102.s04	General Info	Common Criteria requirements
102.s05	General Info	Common Criteria requirements
102.s06	General Info	Common Criteria requirements
102.s07	General Info	Common Criteria requirements
102.s08	Requirement	TCB - provide trusted path security related operations conducted between TCBE and TCB
102.s09	Specification	TCB-to-TCBE Connection Protocol - supports security related operations conducted between the TCB and TCBE
103.s01	Specification	TCB - maintain trusted database server which maintains unique information pertinent to all MLS LAN sessions established on the network
103.s02	Specification	TCB Extension Server - uses Session Status Protocol to change or modify the Session Database Server
103.s03	Specification	TCB - after session establishment, user will be authorized to conduct "normal" operations within the MLS LAN
103.s04	Specification	TCB - normal activity includes "connectivity to the Network Application Protocol Services"
103.s05	Specification	TCB - Application service requests from users are handled by the Secure Session Server
103.s06	Specification	Secure Session Server (SSS) - will validate users session sensitivity level and access
103.s07	Specification	Secure Session Server (SSS) - validates user authorization and creates socket interface if user is authorized
104.s01	Specification	Secure Session Server (SSS) - requires connection protocol that ensures user is presented services commensurate with the current session established by the TCB
104.s02	Specification	TCBE-to-Session Server Protocol - ensures user is presented services commensurate with the current session established by the TCB
104.s03	ISSUE	Application Protocol Server (APS) - must be able to validate client's current session sensitivity level and service authorization
104.s04	Specification	Secure Session Server (SSS) - allows application operations only after validation process
104.s05	Specification	Secure Session Server (SSS) - communicates with the TCB Session Status Database, to compare user service request and user security information maintained by the TCB
104.s06	ISSUE	Client Application Services Validation Protocol - used by SSS to validate user services requests against TCB user security information
105.s00	Figures	Document structure

IU Number	Classification	Description
106.s01	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
106.s02	Future	Non-TCBE-Equipped Workstations Access to Application Protocol Servers
106.s03	Specification	Protected Communications Channel (PCC) - provides two-way hardware identification and authentication between two TCB entities prior to the establishment of trusted path communications the trusted communications
106.s04	Specification	Protected Communications Channel (PCC) - protect all data transmitted between MLS LAN entities, through encryption and verification
106.s05	Specification	All Connection protocols shall only be initiated following establishment of a PCC between the two MLS LAN entities
106.s06	Specification	TCB-to-TCBE Connection Protocol - only initiated through request for "secure attention" from the user
106.s07	Specification	TCB-to-TCBE Connection Protocol - support trusted path security related operations necessary to establish initial session such as login and user identification and authentication, "OR for any user specified operations (logout, set session level, etc.)"""T
106.s08	Specification	TCB-to-TCBE Connection Protocol - only establish session only after activation by the user
106.s09	Specification	TCB-to-TCBE Connection Protocol - shall control the actions of the TCBE through TCBE state commands
106.s10	Future	Session Domination Algorithm
107.s01	Specification	Session Status Protocol (SSP) - initiated for every instantiation or modification of user current session status
107.s02	Specification	Session Status Protocol (SSP) - support trusted communications between TCB Extension Server and Session Database Server
107.s03	Specification	Session Status Protocol (SSP) - support encapsulation of session information
107.s04	Specification	TCBE-to-Session Server Protocol - only initiated following establishment of an Authorized Session between client workstation and the TCB
107.s05	Specification	TCBE-to-Session Server Protocol - support encapsulation of client information necessary for the identification and validation of the user's session sensitivity level and application service request
107.s06	Specification	TCBE-to-Session Server Protocol - allow communications between client and MLS LAN Application Protocol Server only after positive validation of the user's session sensitivity level and the authorization for the specific application service
107.s07	Future	Document structure
108.s00	Document Structure	Blank
109.s00	Reiteration	093.s00
110.s00	Reiteration	094.s00
111.s00	Document Structure	References
112.s00	Document Structure	Blank
113.s01	General Info	Document structure
113.s02	General Info	Document structure
113.s03	General Info	Document structure
114.s00	Document Structure	Title Page
115.s01	Reiteration	113.s01

IU Number	Classification	Description
115.s02	Reiteration	113.s02
115.s03	Reiteration	113.s03
115.s04	General Info	Distribution of memo at the discretion of Dr. Irvine
115.s05	General Info	Document structure
115.s06	General Info	Document structure
116.s00	Document Structure	Table of Contents
117.s01	General Info	Document structure
117.s02	General Info	Document structure
117.s03	General Info	This document does not address all aspects of the MLS LAN architecture
117.s04	General Info	Subsequent documents and established RFCs will address the architectural details of a more advanced nature
117.s05	DEFINITION	Keywords are defined by RFC 2119
117.s06	DEFINITION	Application Protocol Server (APS) - an untrusted, industry standard application protocol server that provides higher layer application services to MLS LAN users
117.s07	DEFINITION	Multilevel Secure (MLS) - Computer system[s] containing information with different sensitivities that simultaneously permits access by users with different security clearances and need-to-know, but prevents users from obtaining access to information for w
117.s08	DEFINITION	Naval Postgraduate School (NPS)
117.s09	DEFINITION	Protected Communications Channel (PCC) - An IPSec secured conduit through which all other MLS LAN connection protocols operate
117.s10	DEFINITION	Protected Channel Initiator (PCI) - A trusted process within the network layer of MLS LAN high assurance servers and TCBE's that provides security services to create a Protected Communications Channel
117.s11	DEFINITION	Secure Attention Key (SAK) - A specified key[s] that when activated will cause a TCBE-equipped MLS LAN workstation to disconnect with all untrusted applications and connect to the TCB
117.s12	DEFINITION	Session Database Server (SDS) - A trusted process within the MLS LAN TCB that manages the session status data for all users logged into the MLS LAN
117.s13	DEFINITION	Secure Session Server (SSS) - A trusted process within the MLS LAN TCB that provides connectivity for users to Application Protocol Servers
117.s14	DEFINITION	Trusted Computing Base (TCB) - A trusted computing base is the collection of security-related elements of a computer system that is responsible for enforcing a security policy
117.s15	DEFINITION	Trusted Computing Base Extension (TCBE) - A high assurance enhanced network interface card (NIC) that is installed into the MLS LAN workstation to support the extension of the TCB
118.s01	DEFINITION	TCB Extension Server - A trusted process within the MLS LAN TCB that conducts the user identification and authentication (I&A) and session negotiation necessary to access the MLS LAN
118.s02	DEFINITION	Workstation - A commercially available personal computer
118.s03	General Info	Additional information in other documents
118.s04	General Info	PCC is based on IPSec - some additional info about IPSec is available in documents listed here
119.s01	Specification	Protected Communications Channel (PCC) - used to establish a conduit through which all other MLS LAN protocols must operate

IU Number	Classification	Description
119.s02	Specification	Protected Communications Channel (PCC) - created through IP layer security as defined by IP Security Standard for the Internet
119.s03	Specification	Protected Communications Channel (PCC) - provides "two-way" mutual hardware authentication between two entities and provides security and integrity protection on all transmitted data
119.s04	Specification	Protected Communications Channel (PCC) - provides some fault tolerance, component loss results in lost communications between the two PCC connected entities but the overall network will not be affected
119.s05	General Info	Document structure
119.s06	General Info	IPSec
119.s07	ISSUE	The specific design of the PCI and data structures necessary for IPSec implementation in the MLS LAN have yet to be finalized
119.s08	ISSUE	The subsequent sections will, provide an approach to be taken in the application of the IPSec in the MLS LAN to create a PCC
119.s09	General Info	IPSec
119.s10	General Info	IPSec
119.s11	General Info	IPSec
119.s12	General Info	IPSec
119.s13	Reiteration	TCBE - using Intel, MLS LAN - uses XTS-300 Server and prototype TCBE using Intel i960 processor - 084.s11
119.s14	Specification	MLS LAN - implement IPSec in a BITS configuration and create PCI as user defined trusted code to be controlled by the security kernel
120.s01	Requirement	MLS LAN - each connection must be encrypted with an algorithm suitable to protect the transmitted information
120.s02	Requirement	Security Manager - responsible for ensuring strength of assigned encryption mechanisms are sufficient to protect given sensitivity level
120.s03	Specification	TCB - maintain virtual table that maps available encryption transforms with the sensitivity levels they can support
120.s04	Specification	Encrypted data is considered to be "safe" for transmission across any medium
120.s05	Specification	Decryption transforms information into a sensitive form
120.s06	Specification	IPSec - provides a mechanism through the Security Policy Database and Security Association Database to segregate the application of protection based upon a set of given attributes[RFC2401]
120.s07	Specification	MLS LAN - Security Manager - create a listing of specific security parameters that a PCC must enforce for connection to each of the MLS LAN entities
120.s08	Specification	MLS LAN - Security Manager - created listings will be mapped to the listing of available MLS LAN session levels enabling the TCB Extension Server to know the Security Policy Database (SPD) assignments for each session level
120.s09	Specification	TCBE - has initial Security Policy Database (Internal and established by the Security manager), only allows connection to the TCB Extension Server
120.s10	Specification	TCB Extension Server - will update the TCBE SPD with the security connection information commensurate with the sensitivity level negotiated for the session

IU Number	Classification	Description
120.s11	Specification	TCBE - using its internal Security Policy Database will correctly negotiate all other connections to the MLS LAN using standard Security Association setup of ISAKMP
120.s12	Future	Additional Encryption Algorithms or transfers
120.s13	Specification	This remote management of the security policy of IPSec is available only because the MLS LAN TCBE can create the initial PCC at system high through the non-volatile Security Policy Database placed on the TCBE
120.s14	Future	TCBE-equipped Workstation treated as non-MLS LAN workstation
120.s15	Future	TCBE-equipped Workstation treated as non-MLS LAN workstation
120.s16	Future	TCBE-equipped Workstation treated as non-MLS LAN workstation
121.s01	Specification	Protected Communications Channel (PCC) - will use standard Internet Key Exchange (IKE) to define a key exchange and to negotiate security services to be provided for each PCC
121.s02	General Info	IKE DOI
121.s03	General Info	IKE DOI
121.s04	ISSUE	current DOI may be sufficient for the MLS environment, but this assumption may be false
121.s05	Specification	Protected Communications Channel (PCC) - first connection must be between the TCBE and TCB Extension Server
121.s06	Specification	Protected Communications Channel (PCC) - initiated by the TCBE once user requests attention from the TCB by activating SAK
121.s07	Specification	Protected Channel Initiator (PCI) - process on the TCBE, will use the initial Security Policy Database setting to establish the IKE phase one exchanges and establish a secure and authenticated communications channel between the TCBE and the TCB Extension
121.s08	Specification	Protected Channel Initiator (PCI) - once IKE security association (SA) has been established, phase two negotiations con then be sent to generate the appropriate incoming and outgoing IPSec Sass
121.s09	Specification	Protected Channel Initiator - will effectively negotiate the specific AH and ESP selectors required for each SA
121.s10	Specification	Protected Communications Channel (PCC) - Each entity will record the SA information into its Security association Database under a unique Security Parameter Index
121.s11	Specification	Protected Communications Channel (PCC) - must be established before the user is allowed to login and negotiate a session.
121.s12	Specification	TCB-TCBE Connection Prtocol - TCB Extension Server - If the session establishment is successful, the TCB Extension Server will issue a "PCC Update" command and transfer the appropriate session level security Policy data to the TCBE for inclusion in its Se
121.s13	General Info	Document structure
121.s14	Specification	PPC - User Requests Application protocol services - the TCBE PCI attempts to create a separate PCC to the source host that supports the requested application protocol server
122.s00	Document Structure	Blank
123.s01	Reiteration	055.s09
123.s02	Reiteration	056.s01
123.s03	Reiteration	056.s02
123.s04	Reiteration	056.s03

IU Number	Classification	Description
123.s05	Reiteration	056.s04
123.s06	Reiteration	056.s05
123.s07	Reiteration	056.s06
123.s08	Reiteration	056.s07
123.s09	Specification	TCBE has 3 state variables
123.s10	Specification	TCBE Power variable - binary, reflects power state of the system
123.s11	Specification	TCBE Trusted Path Operations - binary? Reflects connectivity with TCB and negotiation of a secure session.
123.s12	Specification	TCBE Client OS Loaded variable - binary? Client memory has been purged and "fresh" OS has been loaded
123.s13	Specification	TCBE has 8 total possible states
123.s14	Specification	TCB-TCBE Connection Protocol - TCBE state flag abbreviation: Power = "Power"
123.s15	DEFINITION	Trusted Path Operations (TPO) - this is an abbreviation used in the TCBE state flags
123.s16	Specification	TCB-TCBE Connection Protocol - TCBE state flag abbreviation OS = "Client OS Loaded"
124.s01	Specification	TCBE has disallowed states
124.s02	Specification	TCBE - any state that has Power=Off and any other state variable=YES is disallowed
124.s03	Specification	TCBE - has 3 disallowed states
124.s04	Specification	TCBE - state Power=Off, TPO=Yes, OS=No is disallowed
124.s05	Specification	TCBE - state Power=Off, TPO=No, OS=yes is disallowed
124.s06	Specification	TCBE - state Power=Off, TPO=Yes, OS=Yes is disallowed
124.s07	Specification	TCBE - has 5 allowed states
124.s08	Specification	TCBE - state [2] ?Unprotected Operations? User "Secure Attention Key" starts the "login" Process
124.s08	Specification	TCBE - state [2] trans to state [4] after successful user login
124.s09	Specification	TCBE - state [2] trans to state [2] after UNSUCCESSFUL user login
124.s10	Document Structure	Document Structure
124.s11	Future	Allow the login at "system low" without purge of OS
124.s12	Future	Example of use of 124.s11
124.s13	Specification	TCBE - state [0] Power=Off, TPO=No, OS=No is allowed, this state is named "Power Off"
124.s14	Specification	TCBE - state [1] Power=On, TPO=No, OS=No is allowed, this state is named "Idle"
124.s15	Specification	TCBE - state [2] Power=On, TPO=No, OS=Yes is allowed, this state is named "Untrusted Operations"
124.s16	Specification	TCBE - state [3] Power=On, TPO=Yes, OS=No is allowed, this state is named "Trusted Processing"
124.s17	Specification	TCBE - state [4] Power=On, TPO=Yes, OS=Yes is allowed, this state is named "Trusted Session"
125.s01	Specification	TCB Extension Server - states - uses "response payload type" from TCBE to change "configuration"
125.s01	Specification	TCB Extension Server - states - uses SAR to change "configuration"
125.s02	Specification	TCB Extension Server - state = "configuration"
125.s03	General Info	Document structure
125.s04	Specification	TCB Extension Server - has 5 state variables
125.s05	Specification	TCB Extension Server - Power variable - binary, reflects power state of the system

IU Number	Classification	Description
125.s06	Specification	TCB Extension Server - Connected to TCBE variable - logical connectivity with the TCBE
125.s07	Specification	TCB Extension Server - User Logged in variable - User has successfully completed I&A within the TCB
125.s08	Specification	TCB Extension Server - Session Operations variable - User has successfully negotiated a session security level
125.s09	Specification	TCB Extension Server - Level Changed variable - User has changed his session level
125.s10	Specification	TCB Extension Server - states possible = 32
126.s01	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state flag abbreviation: Power = "Power"
126.s02	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state flag abbreviation Connect = "Connected to the TCBE"
126.s03	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state flag abbreviation Log = "User Logged in"
126.s04	Specification	Session = TCB Extension Server state flag abbreviation "Session Operations"
126.s05	Specification	TCB-TCBE Connection Protocol - TCB Extension Server state flag abbreviation Level = "Level Change"
126.s06	Specification	TCB Extension Server - states disallowed - There is no transition into the disallowed states
126.s07	Specification	TCB Extension Server - stats disallowed - All states that have a Power=No and any other combination of variables =YES are disallowed (15 total)
126.s08	General Info	Document structure
126.s09	Specification	TCB Extension Server - states disallowed - there are a total of 26 disallowed states
126.s10	Specification	TCB Extension Server - states disallowed - any state that has Power=ON Connect=No and any other combination of flags set to yes.
126.s11	Specification	TCB Extension Server - states disallowed - any state that has Power=ON Connect=Yes Log=No and any other combination of flags set to yes.
126.s12	Specification	TCB Extension Server - states disallowed - any state that has Power=ON, Connect=Yes, Log=Yes, Session=No, Level=Yes
127.s01	Specification	TCB Extension Server - states allowed - there are 32 total possible states, 26 are disallowed
127.s02	Specification	TCB Extension Server - states allowed - there are 6 allowed states
127.s03	Document Structure	Document Structure
127.s04	Specification	TCB Extension Server - state [0] Power Off - Power=Off, Connection=No, Log=No, Session=No, Level=No,
127.s05	Specification	TCB Extension Server - state [1] Idle - Power=On, Connection=No, Log=No, Session=No, Level=No,
127.s06	Specification	TCB Extension Server - state [2] Connected - Power=On, Connection=Yes, Log=No, Session=No, Level=No,
127.s07	Specification	TCB Extension Server - state [3] Logged on - Power=On, Connection=Yes, Log=Yes, Session=No, Level=No,
127.s08	Specification	TCB Extension Server - state [4] Running - Power=On, Connection=Yes, Log=Yes, Session=Yes, Level=No,

IU Number	Classification	Description
127.s09	Specification	TCB Extension Server - state [5] Trusted Session Processing - Power=On, Connection=Yes, Log=Yes, Session=Yes, Level=Yes
127.s10	Specification	TCB-to-TCBE Connection Protocol - There are two datagram formats
127.s11	Reiteration	058.s01
127.s12	Specification	Command Datagram - used by the TCB Extension Server to control the TCBE and send information.
127.s13	General Info	Document structure
127.s14	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - all fields are mandatory
128.s00	Figures	FIGURE
129.s01	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - TCB Identifier Header - 32-bit value that identifies the TCBE that created the packet
129.s02	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol -TCB - uses TCB Identifier Header to "facilitate Hardware Identification"
129.s03	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Version Number Field, 4-bit value
129.s04	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Version 1
129.s05	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Payload type field, 4-bit value
129.s06	General Info	Document structure
129.s07	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - three payload types are defined:1. Secure Attention Request, 2. Response, 3. "PCC updated"
129.s08	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Payload length Field, 16-bit
129.s09	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Reserved field, 16-bit, for future use but implented as zeros now
129.s10	Future	FIGURE
129.s11	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Payload field, variable number of 32-bit words.
129.s12	Specification	TCBE-to-TCB Extension Server Datagram (Payload Datagram) Connection protocol - Payload field, pad info to end of 32-bit word
129.s13	General Info	Document structure
129.s14	Specification	TCB Extension Server-to-TCBE datagram (Command datagram)
129.s15	Specification	TCB Extension Server-to-TCBE datagram (Command datagram) - All fields are manditory
130.s01	Specification	TCB Extension Server-to-TCBE datagram (Command datagram) - Response Type, 4-bit field, response type the TCB Extension server expects from the TCBE
130.s02	Specification	TCB Extension Server-to-TCBE datagram (Command datagram) - Response Type, 3 of 16 are presently defined: 0 - No response, 1 - Response with echo, "2 - Response without echo"
130.s03	Specification	TCB Extension Server-to-TCBE datagram (Command datagram) - Command field, 4-bit value, identifies the command the TCB Extension Server is issuing to the TCBE

APPENDIX B: STRAND SPACE FORMALISMS

This appendix presents the process that is used to convert information pertaining to the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols into Strand Space formal specification.[30] This process is presented in 4 sections: The first of these sections, entitled, Protocol Terms, demonstrates how the various protocol message components are represented in the individual protocol terms of the Strand Space formal specification. The second section, entitled Signed Terms, lists the signed terms associated with each authorized participant. The third section, entitled Strands, presents the explicit causatively associated pairs for each of the authorized protocol participants and then shows examples of strands for the authorized participants. The final section, entitled Bundles, presents an example bundle of the three analyzed protocols.

A. PROTOCOL TERMS

The informal protocol descriptions of the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols present a total of five different packet types that may be created by authorized participants.[1] They are presented below, grouped by protocol.

- **TCB-to-TCBE Protocol**
  Payload Packets (Sent from TCBE to the TCB Extension Server)
  Command Packets (Sent from TCB Extension Server to the TCBE)
- **Session Status Protocol**
  Request Packets (Sent from TCB entity to the Session Database Server)
  Response Packets (Sent from Session Database Server to TCB entity)
- **TCBE-to-Session Server Protocol**
  Identification Datagram (Sent from TCBE to Secure Session Server)

1. TCB-to-TCBE Protocol

a. Payload Packets

Payload packets are intended to give the TCBE a way to send information and requests entered by the user to the TCB Extension Server.[1]

Specification	Strand Space term equivalent
TCB Identifier Header	TCB_ID
Version Number	n/a <see note 1 below>
Response Type {0,1,2}	{SAR, Re, PCC_updated}
Payload length	n/a <see note 2 below>
Reserved	n/a <see note 3 below>
Payload	P_x <see note 4 below>

Example composite term: {TCB_ID, SAR, P}

b. Command Packets

Command packets are intended to give the TCB Extension Server a way to send information to the user, via the TCBE, and to direct the actions of the TCBE.[1]

Specification	Strand Space term equivalent
TCB Identifier Header	TCB_ID
Version Number	n/a <see note 1 below>
Response Type {0,1,2}	{NR, RE, RWOE}
Command {0,1,2,3,4,5,6}	{NOOP, RUN, NEW, PCC_UPDATE, RESUME, LOGOUT, DISCONNECT}
Payload length	n/a <see note 2 below>
Reserved	n/a <see note 3 below>
Payload	P_x <see note 4 below>

Example composite term: {TCB_ID, NR, NOOP, P}

2. Session Status Protocol

a. Request Packets

(Sent from TCB entity to the Session Database Server)

Specification	Strand Space term equivalent
TCB Identifier Header	TCBE_ID
User Session ID	TCBE_ID <see note 5 below>
Version Number	n/a <see note 1 below>
Command {0,1,2,3}	{Create, Modify, List, Delete}
Payload length	n/a <see note 2 below>
Reserved	n/a <see note 3 below>
Payload	P_x <see note 4 below>

Example composite term: {TCB_ID, Create, P}

b. Response Packets

(Sent from Session Database Server to TCB entity)

Specification	**Strand Space term equivalent**
TCB Identifier Header	TCBE_ID
User Session ID	TCBE_ID <see note 5 below>
Version Number	n/a <see note 1 below>
Response {0,1,2}	{ACK, NAK, Payload_Response}
Payload length	n/a <see note 2 below>
Reserved	n/a <see note 3 below>
Payload	P_x <see note 4 below>

Example composite term: {TCB_ID, ACK, P}

3. TCBE-to-Session Server Protocol

a. Identification Datagram

(Sent from TCBE to Secure Session Server)

Specification	**Strand Space term equivalent**
TCB Identifier Header	TCBE_ID
TCBE Identification Number	TCBE_ID <see note 6 below>
Version Number	n/a <see note 1 below>
Payload length	n/a <see note 2 below>
Reserved	n/a <see note 3 below>
Payload	n/a <see note 7 below>

Example composite term: {TCB_ID}

Note 1: Version number is a constant in this implementation of the protocols therefore interoperability between protocol versions is not an issue. However, the issue of the interaction of differing version numbers for the protocols will need to be addressed as new versions of the protocols are developed.

Note 2: Payload length is a value that is assumed to be correct in received messages, otherwise they are discarded by the underlying infrastructure, and therefore are not represented in the Strand Space terms.

Note 3: The Reserved field is not used in the present version of the protocols. Therefore, it will not be represented in the Strand Space representations. As

changes are made into the protocols the Reserved field inclusion in future Strand Space representations needs to be re-evaluated.

Note 4: The payload field is represented with a P_x where x is a descriptor of the information in the payload section of the packet. Example: P_SESSION

Note 5: "Version 1 uses the TCBE ID as the User Session ID".[1] Since the information is redundant it will only be represented once in the Strand Space representation.

Note 6: The definition of the TCBE Identification Number is equivalent to the definition of the TCB Identifier Header, since the information is redundant it will only be represented once in the Strand Space representation.

Note 7: "This field is empty in Version 1 of the protocol".[1]

B. SIGNED TERMS

There are four authorized participant roles related to the protocols of interest. They are: the TCBE-equipped workstations, simply referred to as TCBE, the TCB Extension Server, The Session Database Server, and the Secure Session Server

1. TCBE:

The following is a list of signed terms for the TCBE:

+{TCB_ID, SAR, P_undefined},
+{TCB_ID, Re, P_SESSION}
+{TCB_ID, Re, P_SESSION_LEVEL_CHANGE}
+{TCB_ID, Re, P_SET_GROUP}
+{TCB_ID, Re, P_LOGOUT}
+{TCB_ID, Re, P_RUN}
+{TCB_ID, PCC_updated, P_undefined},
+{TCB_ID},
-{TCB_ID, NR, NOOP, P_Session_level_information}
-{TCB_ID, NR, RUN, P_undefined}
-{TCB_ID, NR, LOGOUT, P_undefined}
-{TCB_ID, NR, NOOP, P_disconnect}
-{TCB_ID, RE, NOOP, P_username}
-{TCB_ID, RE, NOOP, P_session_change_level}
-{TCB_ID, RE, NOOP, P_user_interface_menu}

-{TCB_ID, RWOE, NOOP, P_password}
-{TCB_ID, RWOE, PCC_UPDATE, P_undefined}

Assumed Packets:
+{TCB_ID, Re, P_USER}
+{TCB_ID, Re, P_PASSWORD}
+{TCB_ID, List, P_undefined}¥

2. TCB Extension Server:

The following is a list of signed terms for the TCB Extension Server:

+{TCB_ID, NR, NOOP, P_Session_level_information}
+{TCB_ID, NR, RUN, P_undefined}
+{TCB_ID, NR, LOGOUT, P_undefined}
+{TCB_ID, NR, NOOP, P_disconnect}
+{TCB_ID, NR, RESUME,P_undefined}
+{TCB_ID, NR, NEW, P_undefined}
+{TCB_ID, RE, NOOP, P_username}
+{TCB_ID, RE, NOOP, P_session_change_level}
+{TCB_ID, RE, NOOP, P_group_change}
+{TCB_ID, RE, NOOP, P_user_interface_menu}
+{TCB_ID, RWOE, NOOP, P_password}
+{TCB_ID, RWOE, PCC_UPDATE, P_undefined}
+{TCB_ID, Create, P_SSD_info}
+{TCB_ID, Modify, P_SSD_info}
+{TCB_ID, List, P_undefined}
+{TCB_ID, Delete, P_undefined}

-{TCB_ID, SAR, P_undefined},
-{TCB_ID, Re, P_SESSION}
-{TCB_ID, Re, P_SESSION_LEVEL_CHANGE}
-{TCB_ID, Re, P_SET_GROUP}
-{TCB_ID, Re, P_LOGOUT}
-{TCB_ID, Re, P_RUN}
-{TCB_ID, PCC_updated, P_undefined},

-{TCB_ID, Request_TCB_ID, ACK, P_undefined}
-{TCB_ID, Request_TCB_ID, NAK, P_undefined}
-{TCB_ID, Request_TCB_ID, Payload, P_SSD_info}

ASSUMED PACKETS:
-{TCB_ID, Re, P_USER}
-{TCB_ID, Re, P_PASSWORD}

¥ See Extraneous Abilities on page 43

3. Session Database Server:

The following is a list of signed terms for the Session Database Server:

+{TCB_ID, Request_TCB_ID, ACK, P_undefined}
+{TCB_ID, Request_TCB_ID, NAK, P_undefined}
+{TCB_ID, Request_TCB_ID, Payload, P_SSD_info}

-{TCB_ID, Create, P_SSD_info}
-{TCB_ID, Modify, P_SSD_info}
-{TCB_ID, List, P_undefined}
-{TCB_ID, Delete, P_undefined}

4. Secure Session Server:

The following is a list of signed terms for the Session Database Server:
+{TCB_ID, List, P_undefined}

-{TCB_ID}

C. STRANDS

This section presents strand relationship for each of the authorized participants of the protocols. This is presented in two sections. The first section, entitled Associated Pair Listing, presents a listing of each of the explicatively causatively associated pairs by protocol participant. Explicatively causatively associated pairs are pairs that are comprised of a negatively signed term connected to a positively signed term using the => relationship, as shown in Figure 19. The second section, entitled Example Strands, presents a few examples of full stands associated with authorized participants of the protocol.

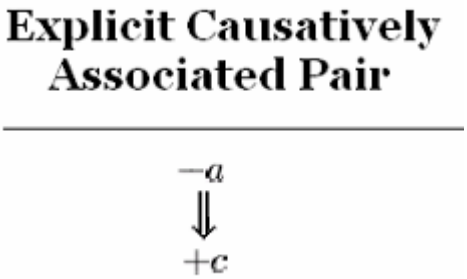


Figure 19. Explicit Causatively Associated Pair

These pairs will be presented in the format presented in Table 18.

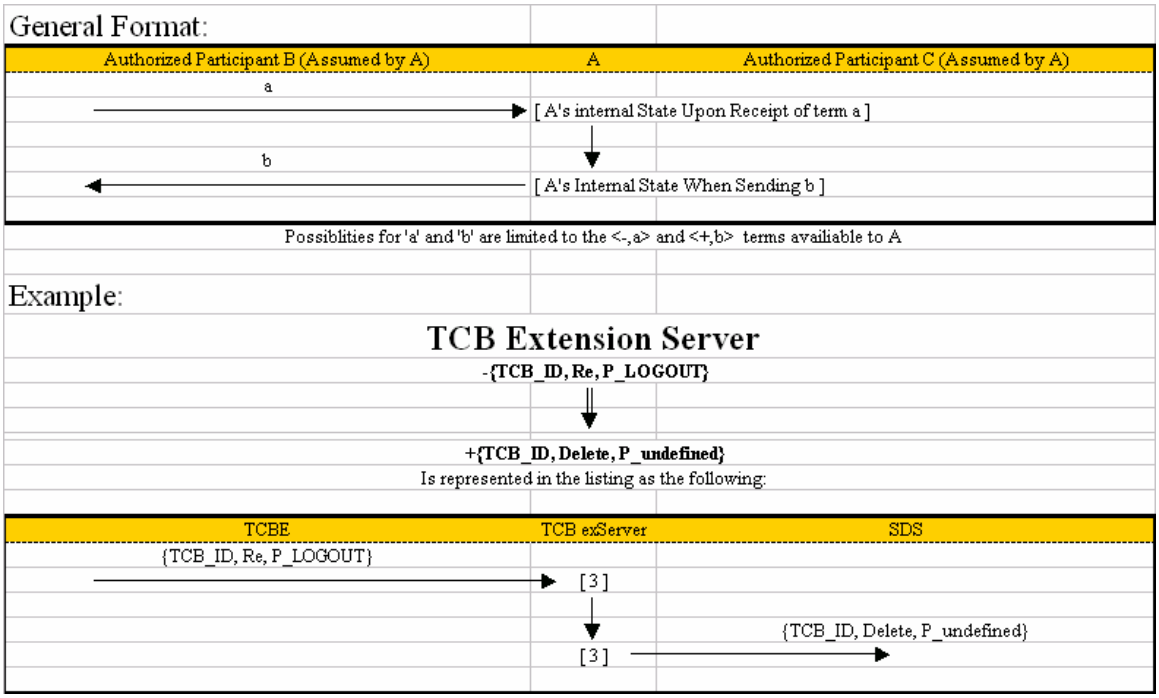


Table 18. Format of Explicit Causative Associated Pair Listing

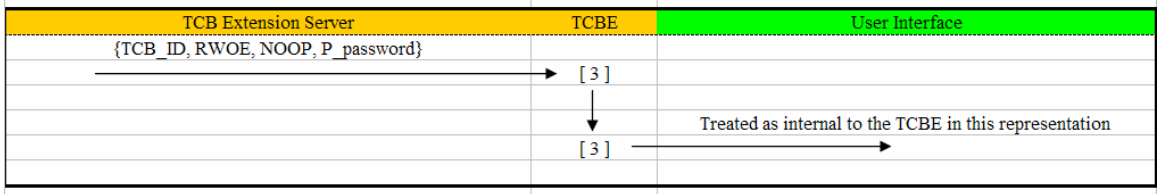
1. Associated Pair Listing

a. TCBE

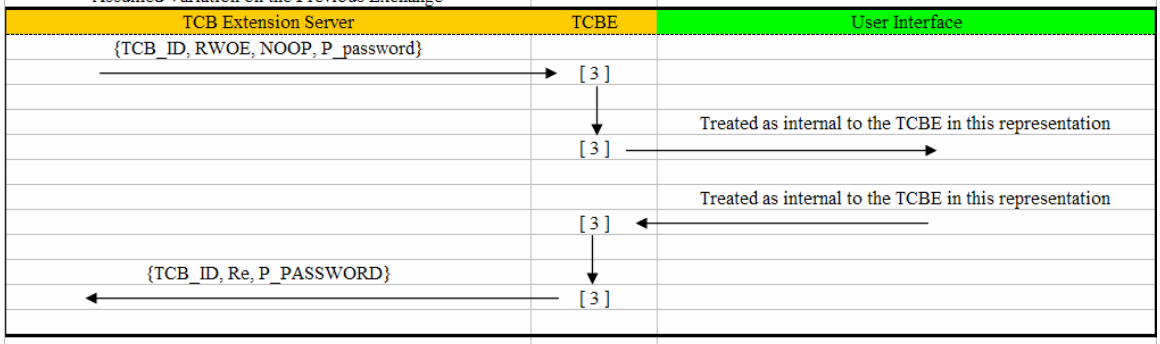

TCBE
Strand Space Notation
Assumed Variation on the Previous Exchange
TCB Extension Server
TCBE
User Interface
{TCB_ID, RE, NOOP, P_session_change_level}
[3]
Treated as internal to the TCBE in this representation
[3]
Treated as internal to the TCBE in this representation
[3]
{TCB_ID, Re, P_SESSION_LEVEL_CHANGE}
[3]
TCB Extension Server
TCBE
User Interface
{TCB_ID, RE, NOOP, P_group_change}
[3]
Treated as internal to the TCBE in this representation
[3]
Assumed Variation on the Previous Exchange
TCB Extension Server
TCBE
User Interface
{TCB_ID, RE, NOOP, P_group_change}
[3]
Treated as internal to the TCBE in this representation
[3]
Treated as internal to the TCBE in this representation
[3]
{TCB_ID, Re, P_SET_GROUP}
[3]
TCB Extension Server
TCBE
User Interface
{TCB_ID, RE, NOOP, P_username}
[3]
Treated as internal to the TCBE in this representation
[3]
Assumed Variation on the Previous Exchange
TCB Extension Server
TCBE
User Interface
{TCB_ID, RE, NOOP, P_username}
[3]
Treated as internal to the TCBE in this representation
[3]
Treated as internal to the TCBE in this representation
[3]
{TCB_ID, Re, P_PASSWORD}
[3]

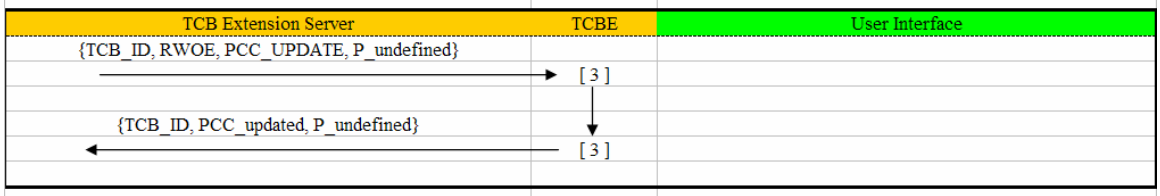
TCBE

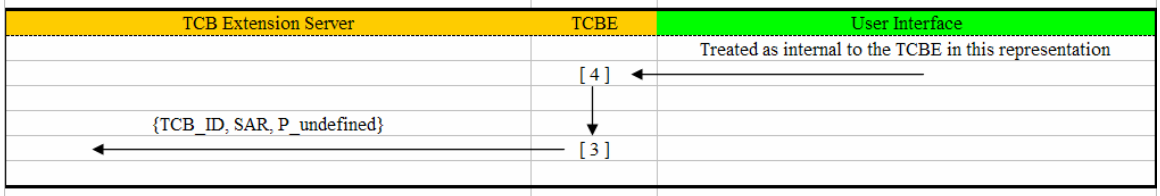
Strand Space Notation

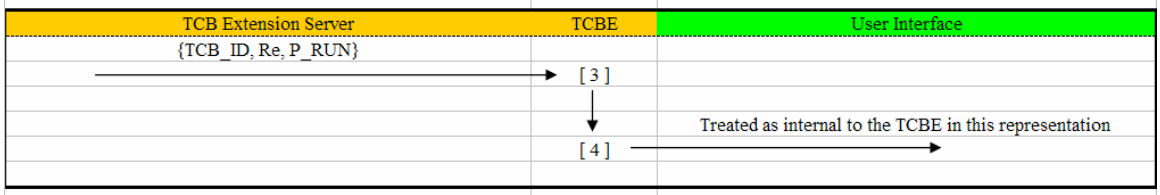


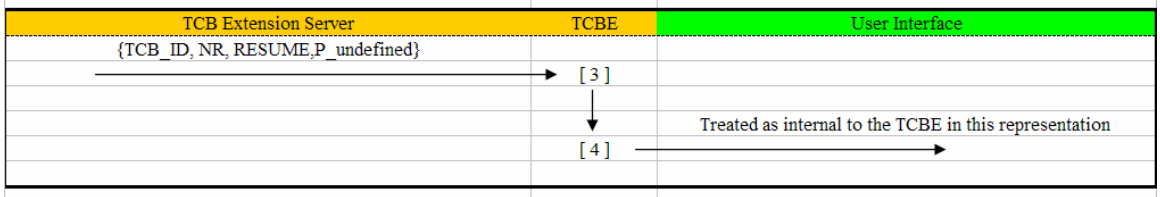
Assumed Variation on the Previous Exchange

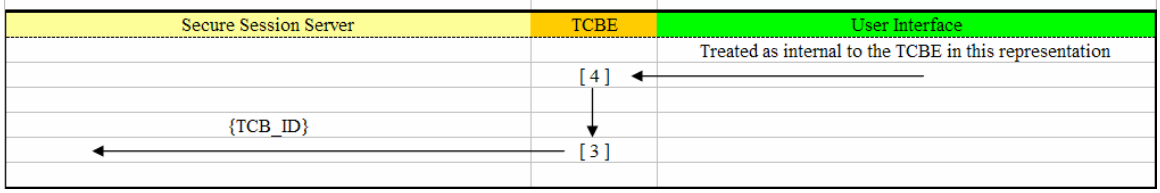




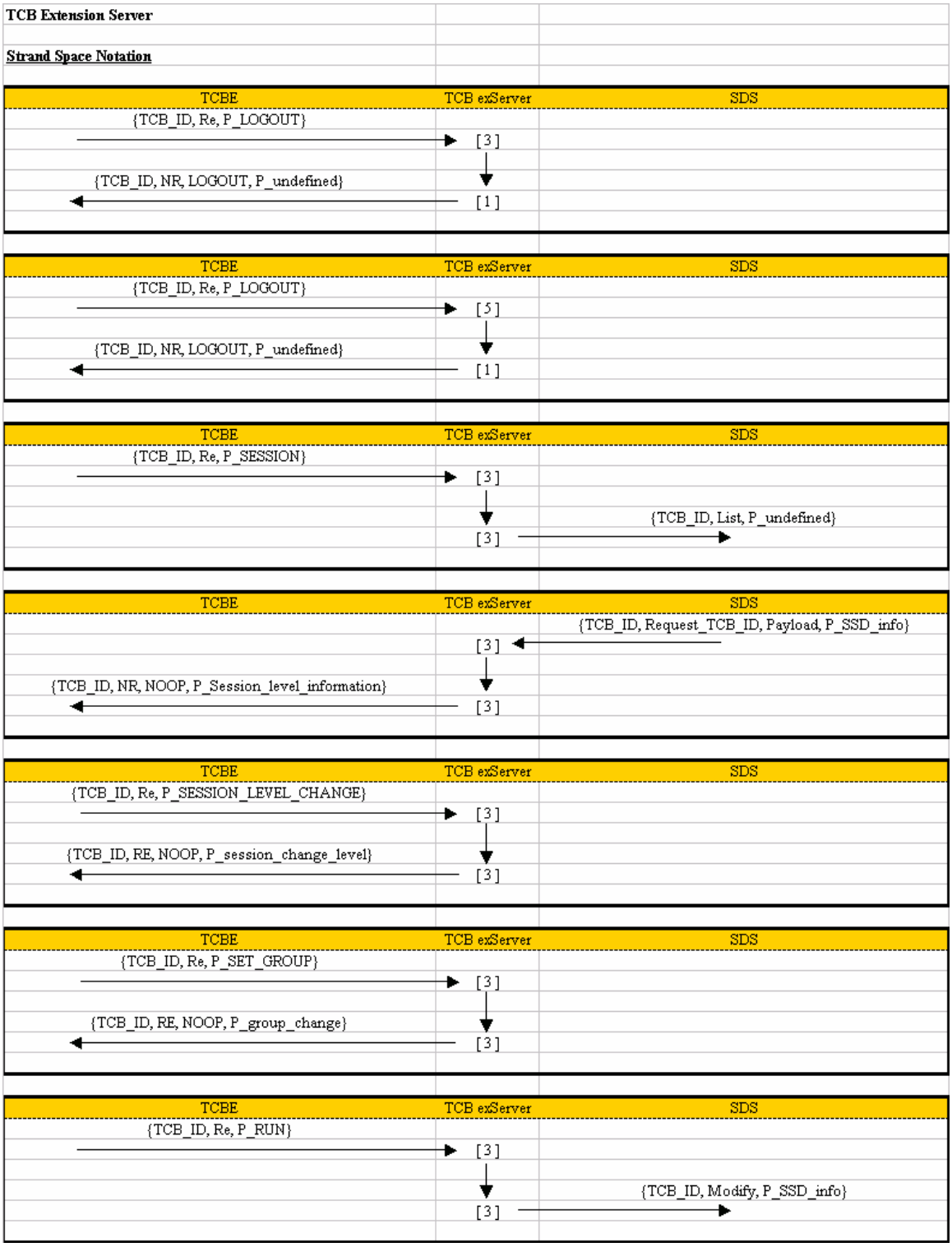






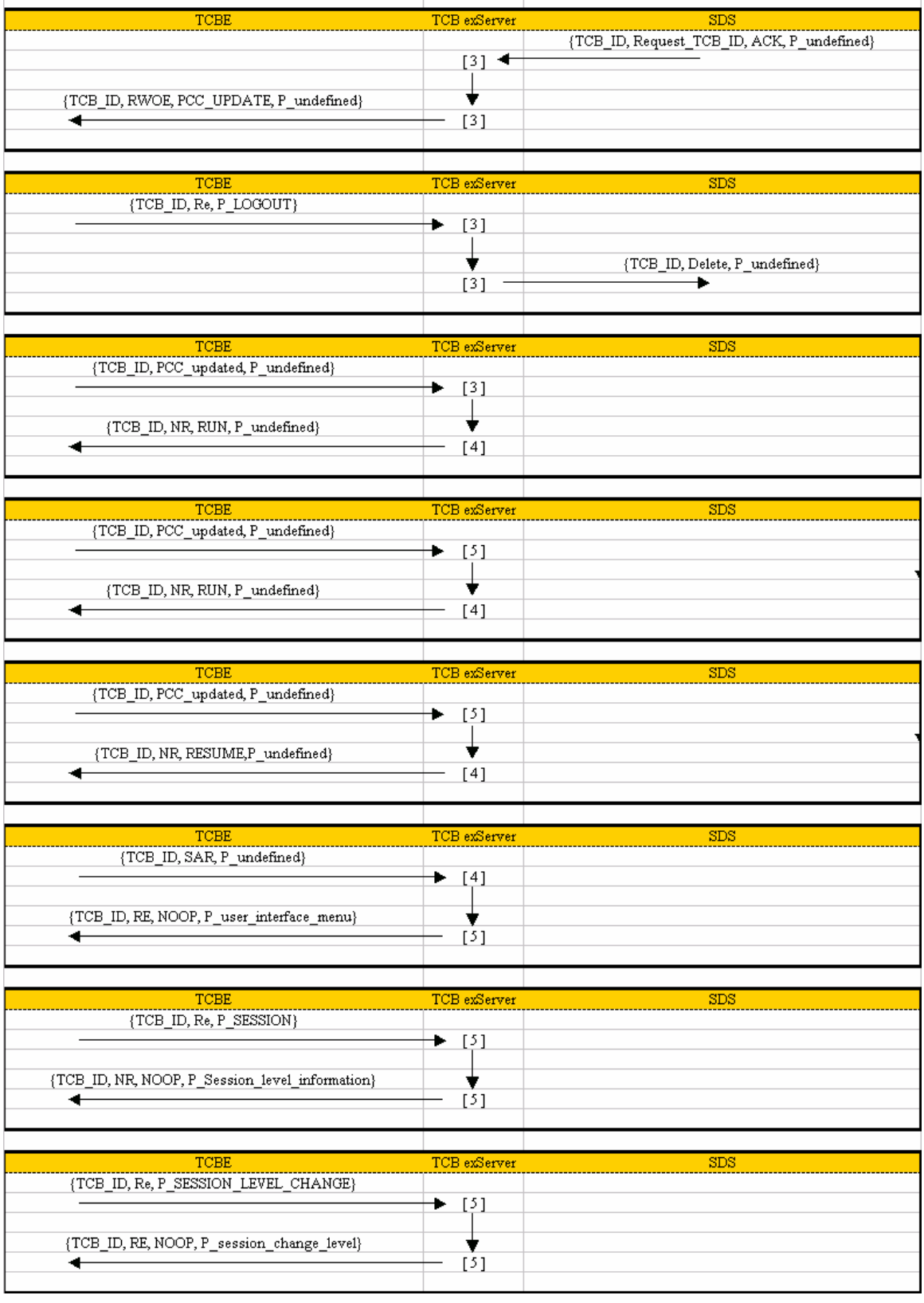


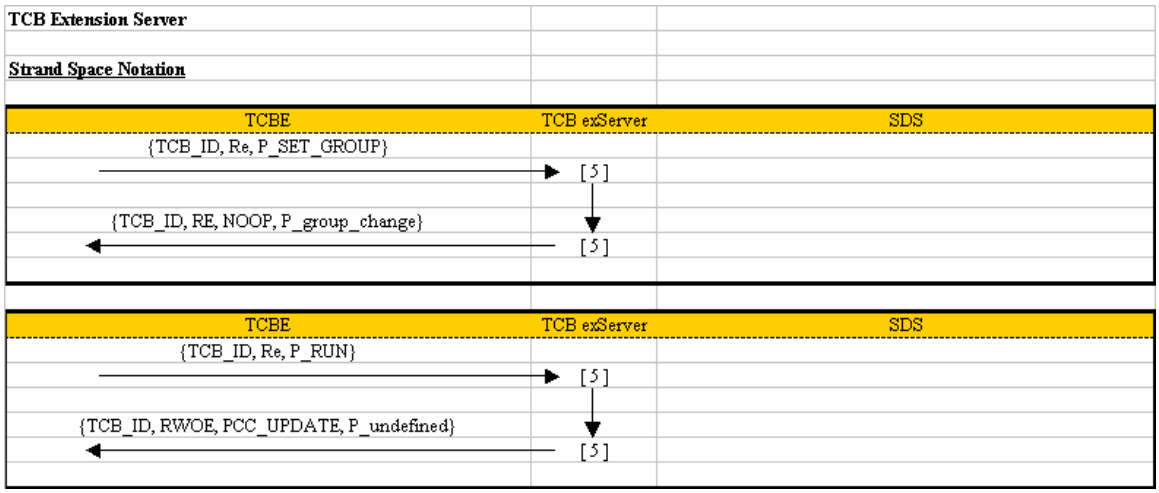
b. TCB Extension Server



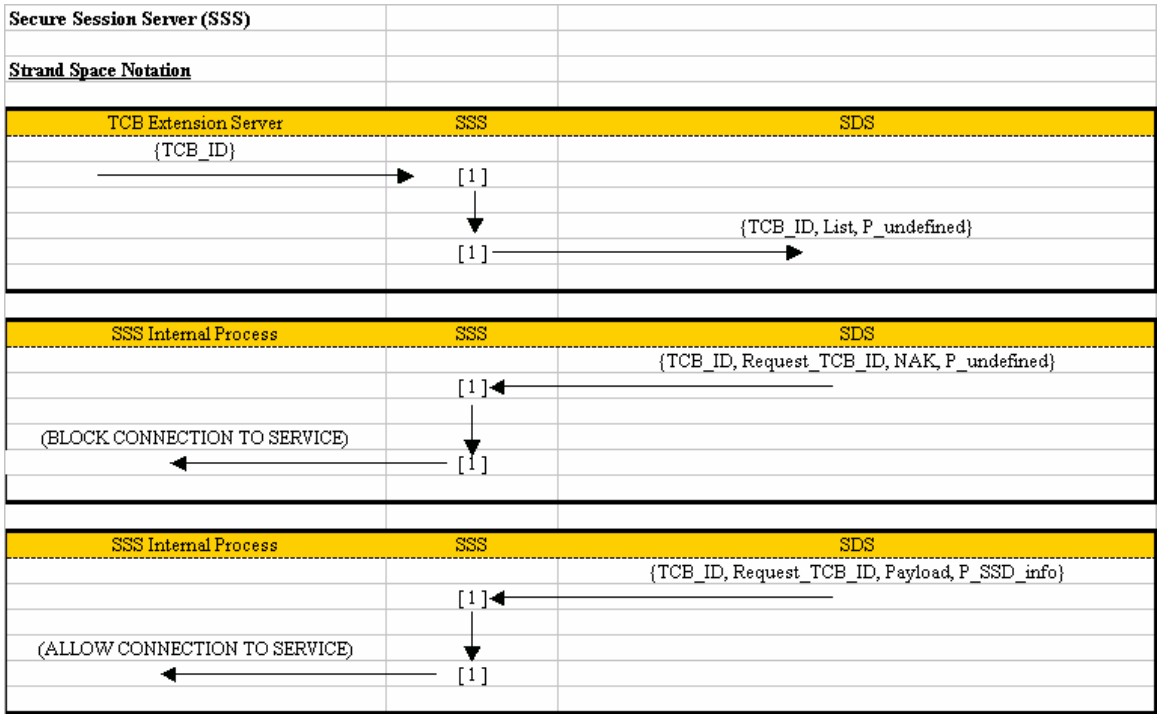
TCB Extension Server

Strand Space Notation





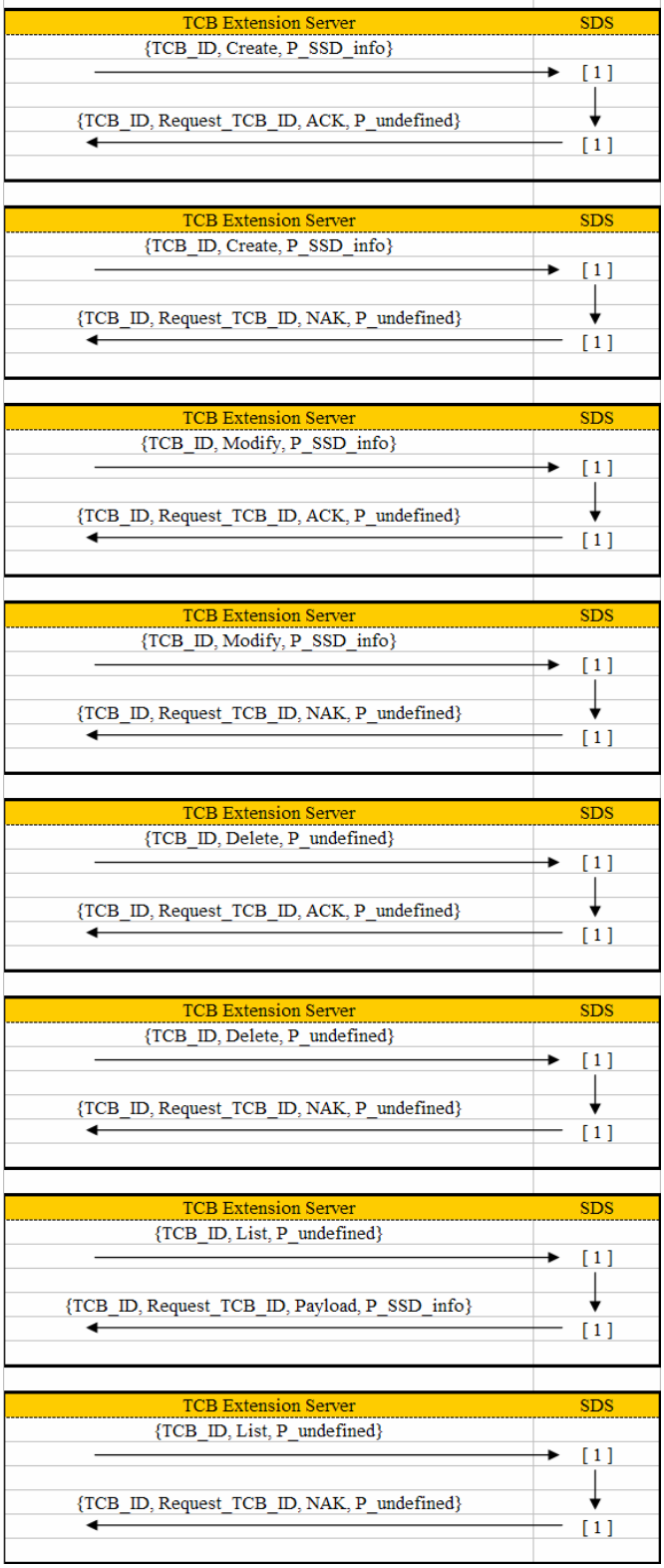
c. Secure Session Server



d. Session Database Server

Secure Database Server (SDS)

Strand Space Notation



2. Example Strands

a. TCBE

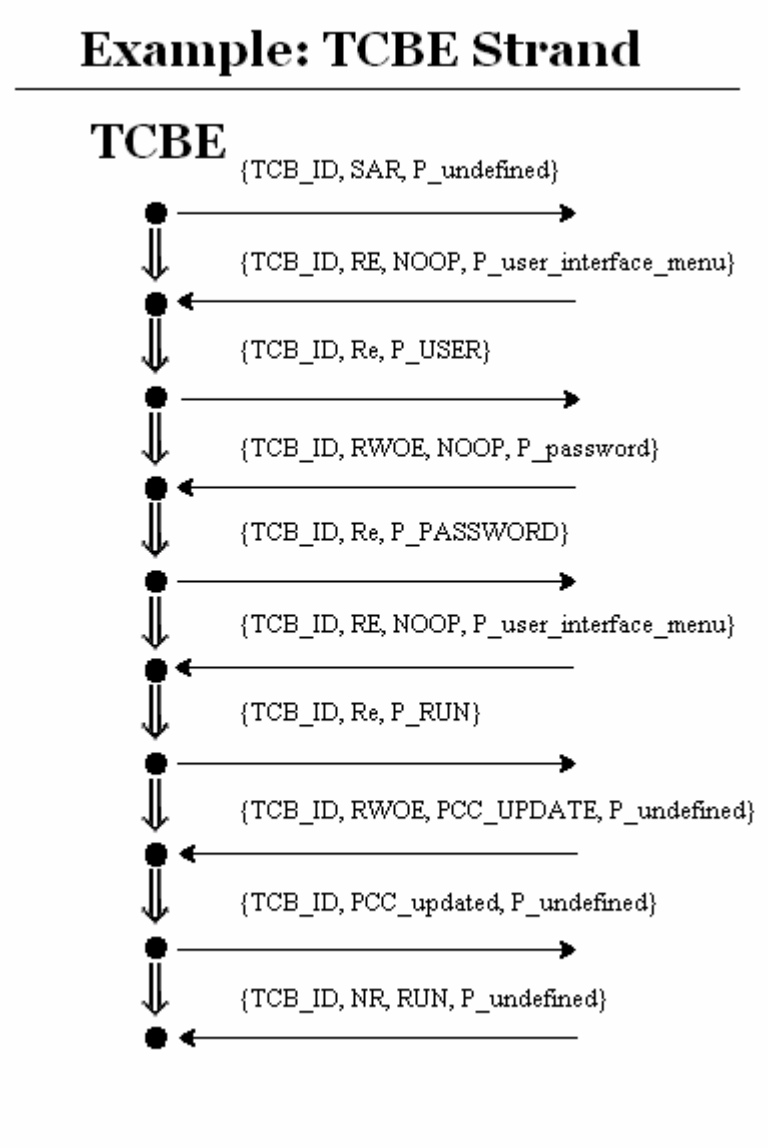


Figure 20. Example of TCBE Strand

b. TCB Extension Server

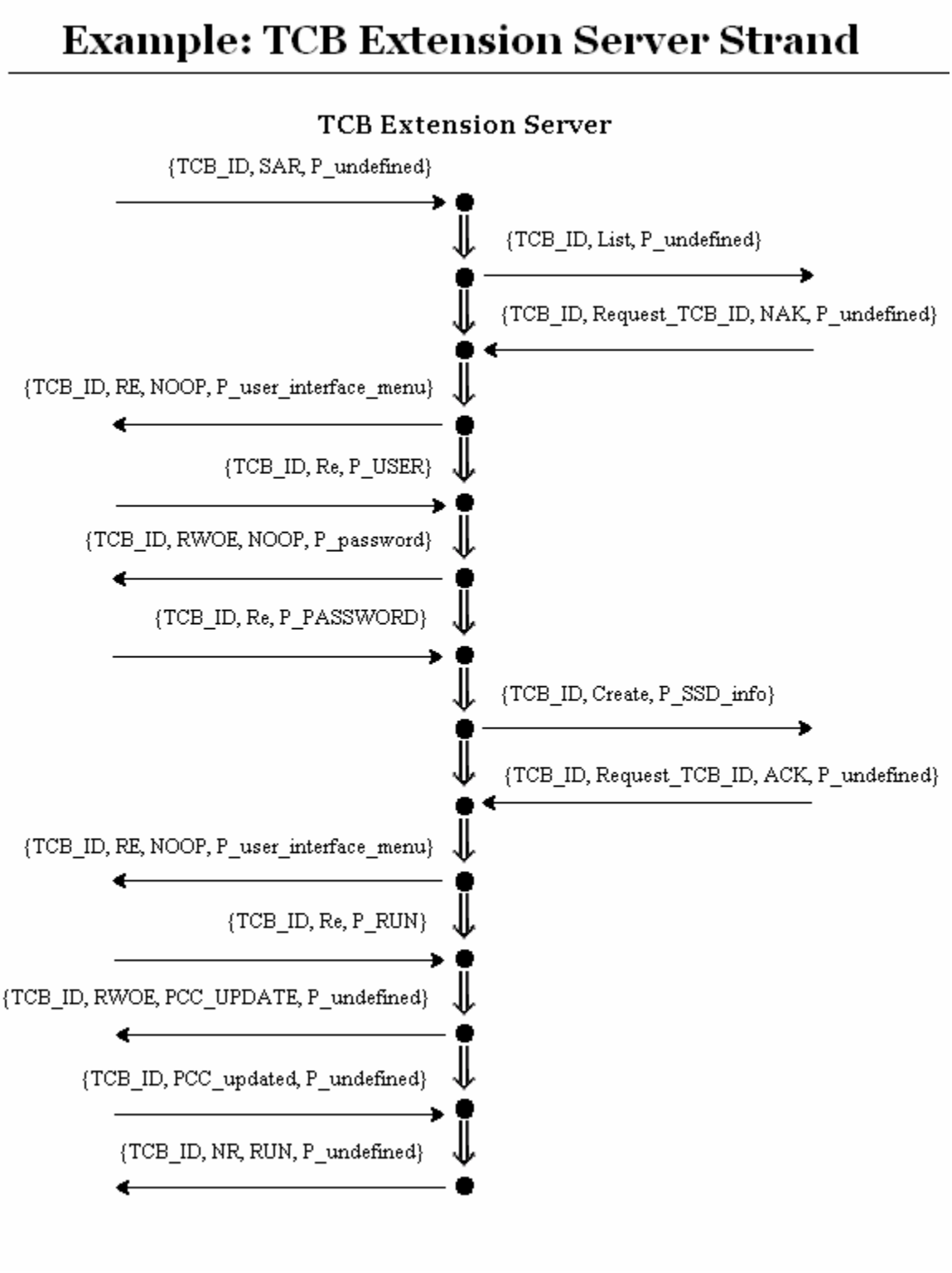


Figure 21. Example of TCB Extension Server Strand

c. Secure Session Server

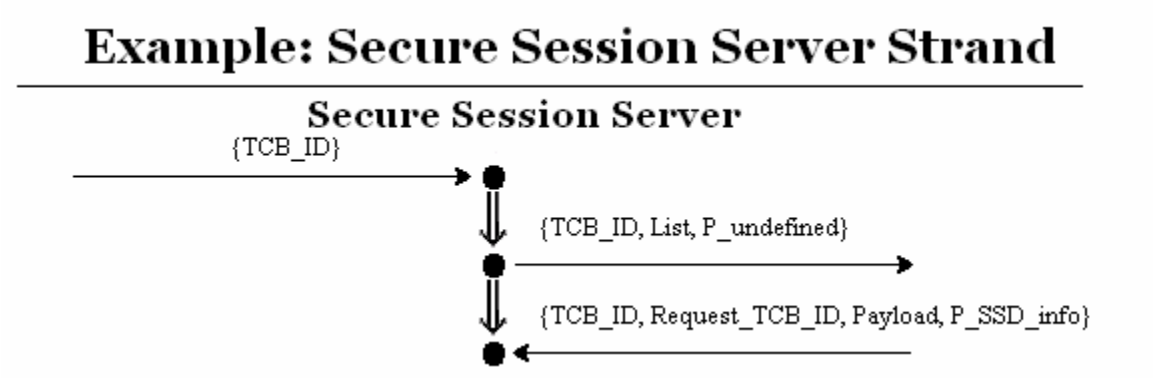


Figure 22. Example of Secure Session Server Strand

d. Session Database Server

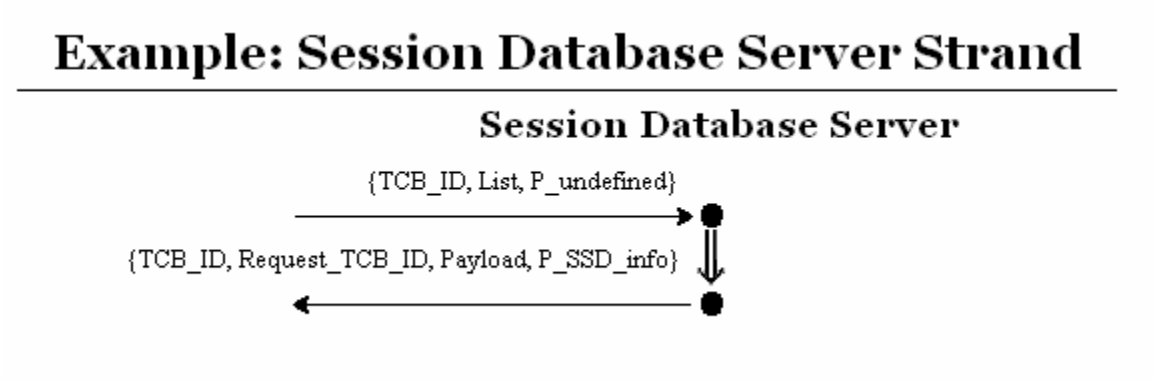


Figure 23. Example of Session Database Server Strand

D. BUNDLES

This section presents an example bundle. A bundle is formed when two or more strands are "connected" using the causal link representation →. This is used to represent that one strand sends a term and the "connected" strand receives an equivalent term. The bundle in Figure 24 presents a bundle that consists of all of the protocols of interest, represented in black. Additionally, user interaction and other assumptions are presented in blue in order to add context to the protocol interactions.

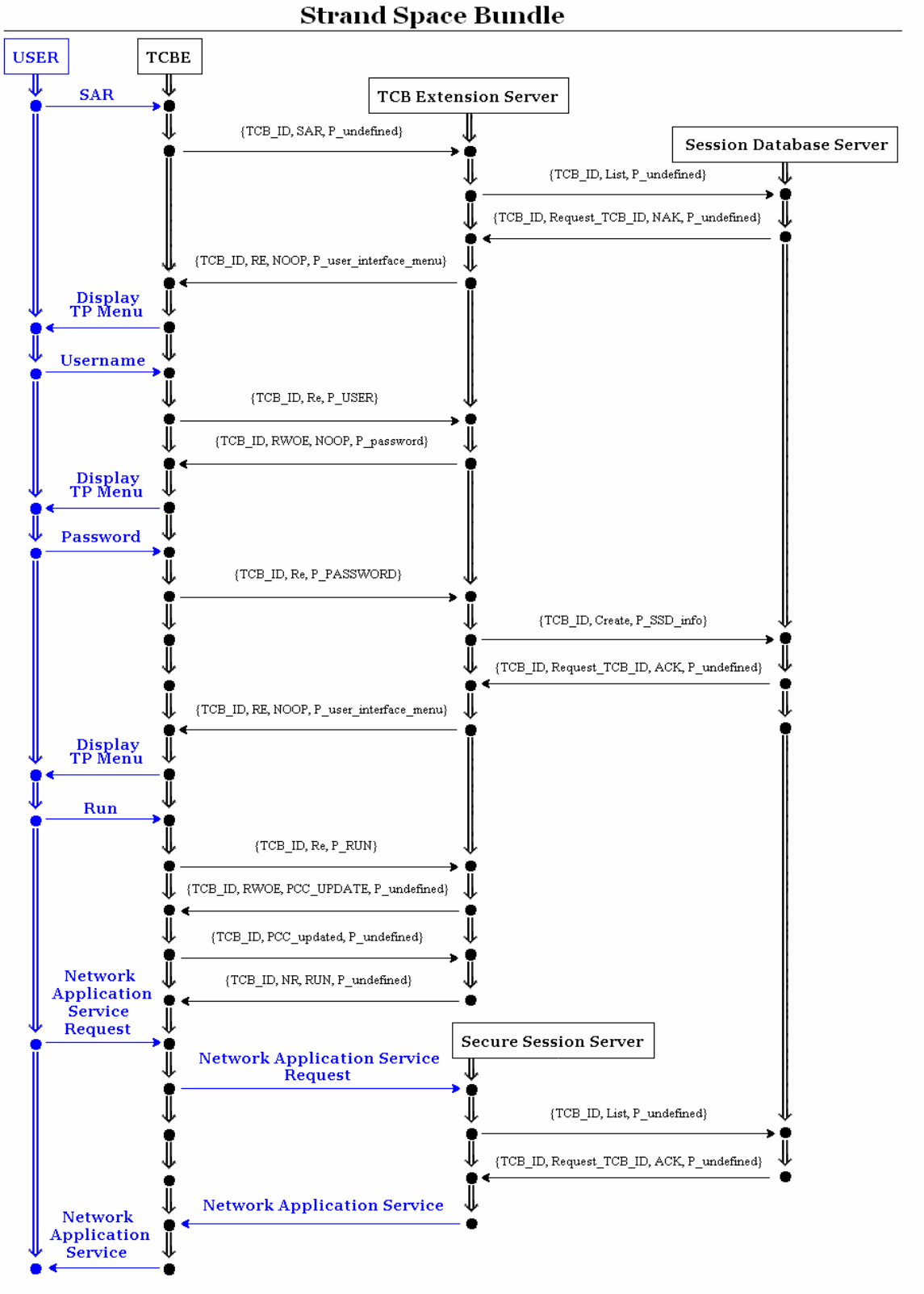


Figure 24. Stand Space Bundle

APPENDIX C

The application of automated tools to cryptographic analysis is a valuable asset. Even though secrecy and mutual authentication, for the TCB-to-TCBE, Session Status, and TCBE-to-Session Server protocols is provided by the Protected Communications Channel, the application of an automated tool still has benefits. Those benefits are two fold. The first of these is by presenting a simple and efficient tool for protocol analysis; one provides the development team with an idea of the time cost benefit of the application of such tools. Secondly, the application of such a tool increases the confidence in the results of the hand analysis completed in appendix B.

The tool that is implemented was developed by John Millen and is simply called the Constraint Solver. This is a natural section for use in this paper because the tool is based on strand space models. The tool is based on the idea that the certain reachability problems for cryptographic protocols can be solved using a constraint satisfaction procedure.[38] The tool is implemented in SWI-Prolog.[39]

A. STEPS IN THE PROCESS

The first step in the process is to create protocol roles. These correspond to the entities of the MLS LAN as well as the penetrator. The second step is to develop a set of tests that have a set number of participants and a specific test term. This analysis focuses on the secrecy properties of the protocols interactions; give the assumptions about the environment. One important note is that the secrecy goal states that some designated messages should not be made public.

B. CODE

1. csolve_pl[39]

(THIS CODE IS FROM http://www.csl.sri.com/users/millen/capsl/constraints.html)

```
/*
Protocol analysis based on "Constraint Solving
for Bounded Cryptographic Protocol Analysis"
ACM CCS-8, 2001

N-ary concatenation, but not associative
   Elements of a cat may be cats
Use search for convenience

'stop' and 'Auth' tests for secrecy and authentication

Operators:

[U,V, ...] is concatenation, n-ary

U*K is U pk-encrypted with K, usually pk(A)

U/pk(A) is signature of U by A (not invertible)

U+K is U encrypted with K as symmetric key

U-K is hidden symmetric encryption (see paper)

sha(U) is a hash function

e is the attacker

msk(A,B) = msk(B,A) mutual (shared) symmetric key
*/

% :- table(solve/2). % for XSB Prolog only

:- dynamic(cc/1).
:- assert(cc(0)).

% resetcc resets the constraint set count to zero
% (use it between trials)
% ics increments it by one, used in reach

resetcc :-
  retractall(cc(_)),
  assert(cc(0)).

ics(N1) :-
  retract(cc(N)),
  N1 is N+1,
  assert(cc(N1)).

search(B,Auth) :- % Typical reach call
  search(B,[a,b,e],Auth).
```

```
search(B,I,Auth) :-
  write('Starting csolve...'),nl,
  resetcc,
  reach(B,[],I,F,[],Lout,Auth),nl,
  write('Simple constraints:'),nl,prlist(F),nl,
  reverse(Lout,Tr),
  write('Trace:'),nl,prlist(Tr),nl,
  write('Bundle:'),nl,prlist(B),nl.

%    reach(Bundle,Constraintlist,Terms,FinalConstraintList,interLeavings
in/out, Auth)
%   Constraintlist initially empty
%     Constraint is [term, termlist]
%   Terms is a list of terms known to attacker
%     Terms initially just principal names
%   Bundle is a list of strands. (Actually a "semibundle")
%     Strand is list of send(M) and recv(M) nodes
%   Interleavings: Lin initially empty, Lout variable
%   Auth is a pattern used for authentication tests
%     Auth=event_name(A1,A2,...)  any event_name OK
%     Auth message sent causes immediate solve failure
%     Auth=[] for no auth. test
% reach creates the initial list of constraints
%   from a possible merge and passes it to solve

reach(B,C,_T,F,Lin,Lin,Auth) :-
  allnull(B),
  ics(N),
  write(' Try '),write(N),
%  prlist(Lin),
  Auth =.. H,!,
  solve(C,F,H).

reach(B,C,T,F,Lin,Lout,Auth) :-
  selectnode(B,send(M),B1),!, % send adds term
  reach(B1,C,[M|T],F,[send(M)|Lin],Lout,Auth).

reach(B,C,T,F,Lin,Lout,Auth) :-
  selectnode(B,recv(M),B1),   % recv adds constraint
  reach(B1,[[M,T]|C],T,F,[recv(M)|Lin],Lout,Auth).

% selectnode(B,N,B1) separates B into the first node
% N of some strand and the remaining strand set B1.
% selectnode fails if B is all null.
% Note: to optimize, we select all send nodes first, any order.
% (but usually only one send is available anyway)
% If all nodes are recv, order does matter, so all orders
% are attempted.

selectnode(B,send(M),B1) :-
  member([send(M)|_S],B),!,     % this cut for send optimization
  diff(B,send(M),B1).

selectnode([[recv(M)|S]|B],recv(M),[S|B]).  %  remove  recv  from  first
strand,

selectnode([S|B],recv(M),[S|B1]) :-  % or from some other strand
```

```
  selectnode(B,recv(M),B1).

% diff(B,N,B1): bundle B minus node N is B1

diff([[N|S]|B],N,[S|B]).

diff([S|B],N,[S|B1]) :- diff(B,N,B1).

% solve([[expr, termlist],...],Varconstraints)
% apply reduct to each nonsimple constraint, in
% reverse (i.e., chronological) order.
% Build up (possibly empty) list of simple [var,termlist] constraints.
% Note that reduct may cause a var to be instantiated
% on the left side of a prior constraint, so
% tail recursion applies solve again from the beginning.

solve(C,C,_) :- allvarc(C),!.

solve([[A,T]|C],W1,H) :-
  solve(C,V,H),
  remv(T,T1),
  solve1(A,T1,C,V,W1,H). % test for stopping conditions

solve1(_A,T,_C,_V,_W,_H) :-
  member(stop,T),!.

solve1(_A,T,_C,_V,_W,H) :-
  authmatch(T,H),!,fail.

solve1(A,T1,C,V,W1,H) :-
  reduct(A,T1,U),
  append(U,V,W),
  solve(W,W1,H).

% reduct(M,T,C) performs one reduction step on
% an active constraint M:T and
% records replacement constraints in C

% "safe" steps preserve all possible solutions

reduct(M,T,[[M,T]]) :-
  var(M),!.       % pass over simple constraint

reduct(M,T,[]) :-       % (un) with constant
  atomic(M),            % always safe
  member(M,T),!.

reduct([A,B],T,[[B,T],[A,T]]) :- !.       % (pair), always safe

reduct([A|C],T,[[A,T]|D]) :- !, % (pair) extended, always safe
  reduct(C,T,D).

reduct(M/pk(e),T,[[M,T]]) :- !.     % (sig), always safe

reduct(M,T,[]) :-       % (un)
  member(A,T),
  hunify(M,A).
```

```
reduct(sha(M),T,[[M,T]]).     % (hash)

reduct(pk(A),T,[[A,T]]).      % public-key lookup

reduct(msk(e,A),T,[]).        % e knows own shared secret keys
reduct(msk(A,e),T,[]).
reduct(csk(e),T,[]).          % unary form of secret key

reduct(M*K,T,[[M,T],[K,T]]).  % (penc)

reduct(M+K,T,[[M,T],[K,T]]).  % (senc)

reduct(M,T,[[M,T]]) :-
  do_ksub(T).

reduct(M,T,[[M,T2],[K,T1]]) :-
  do_ksyn(T,T2,K,T1).

% remv removes variables from a term list, if any.
% It also does (split) and (pdec), they're always safe.

remv([],[]) :- !.

remv([A|T],W) :-
  var(A),!,
  remv(T,W).

remv([[A,B]|T],W) :- !, % (split) for pair
  remv([A,B|T],W).

remv([[A|B]|T],W) :- !, % (split) extended
  remv([A,B|T],W).

remv([U*K|T],W) :-
  K==pk(e),!,           % (pdec)
  remv([U|T],W).

remv([A|T],[A|W]) :-
  remv(T,W).

% do_ksub looks for U*V in a term list
% and binds V to pk(e) if possible (and V not already pk(e))
% It fails if there is no instance to bind.

do_ksub([_U*V|_T]) :-
  \+V==pk(e),
  V=pk(e).

do_ksub([_A|T]) :- do_ksub(T).

% do_ksyn looks for U+K in a term list T
% and decrypts it to U.  We also insert the
% new constraint for K with U-K in T1.
% Fails iff there is no symmetric encryption.

do_ksyn([U+K|T],[U,K|T],K,[U-K|T]).
```

```
do_ksyn([A|T],[A|T2],K,[A|T1]) :-
  do_ksyn(T,T2,K,T1).

% allnull tests for empty bundle

allnull([]).

allnull([[]|B]) :- allnull(B).

% allvarc tests for simple constraint set
%    in which all left sides are variables

allvarc([]).

allvarc([[X,_T]|C]) :-
  var(X),
  allvarc(C).

% hunify(M,A) turns A from - to + first if necessary.
%  (can show that "-" can occur only at top level)

hunify(M,U-V) :- unify(M,U+V),!.
hunify(M,A) :- unify(M,A).

%-------------------------------------------------------------------------
------
% "safe" unification with occurs check from C. Meadows
%-------------------------------------------------------------------------
------

unify(X,Y) :-
      var(X),var(Y),!,
      X=Y.
unify(X,Y) :-
      atomic(X),!,X=Y.  % atomic includes numbers
unify(X,Y) :-
      atomic(Y),!,X=Y.
unify(X,Y) :-
      var(X),!,
      notOccurs(X,Y),
      X=Y.
unify(X,Y) :-
      var(Y),!,
      notOccurs(Y,X),
      X=Y.
unify(X,Y) :-
      X =.. [A|B],
      Y =.. [A|C],
      list_unify(B,C),!.

unify(msk(A,B),msk(B1,A1)) :- % msk is commutative
      unify(A,A1),
      unify(B,B1).

list_unify([],[]).
list_unify([A|B],[C|D]) :-
```

```
      unify(A,C),
      list_unify(B,D).

notOccurs(X,Y) :- var(Y),!, \+ X == Y.
notOccurs(_X,Y) :- atomic(Y),!.
notOccurs(X,[Y|Z]) :- !,notOccurs(X,Y),notOccurs(X,Z).
notOccurs(X,Y) :- Y =.. [_F|N],notOccurs(X,N).

%-------------------------------------------------------------------------
------
%     Printing
%-------------------------------------------------------------------------
------

% Print list elements

prlist([]).
prlist([X|L]) :-
  write(X),nl,
  prlist(L).


%-------------------------------------------------------------------------
------
% authmatch(T,H) finds a pattern match of H to some element of T
%   without binding any variable in T.
%-------------------------------------------------------------------------
------

authmatch(T,[[]]) :- !,fail. % no Auth pattern
authmatch([A|T],H) :-
  A=..AL,
  authmatch1(AL,H),!.
authmatch([A|T],H) :- authmatch(T,H).
authmatch1([],[]).
authmatch1([X|U],[Y|V]) :-
  authmatch1a(X,Y),
  authmatch1(U,V).
authmatch1a(X,Y) :- var(Y),!,Y=X.
authmatch1a(X,Y) :- var(X),!,fail.
authmatch1a(X,Y) :- X==Y,!.
authmatch1a(X,Y) :- atomic(Y),!,fail.
authmatch1a(X,Y) :- atomic(X),!,fail.
authmatch1a(X,Y) :- X=..XL,Y=..YL,authmatch1(XL,YL).
```

2. MLS_LAN_Protocols

```
% MLS_LAN_Protocols
% Written By Daniel Craven
% For use with J. Millen's Constraint Checker found on
% http://www.csl.sri.com/users/millen/capsl/constraints.html

%---------------------------------
%TBCE role is roleA
% shares a symmetric key with the TCB Extension Server
% which is labeled KeyAB
%---------------------------------
strand(roleA,A,B,D,KeyAB,KeyBD,Na,Nb,Nd,[
  recv([A,B,D]),
  send([A,Na,sar]*KeyAB),
  recv([B,Nb,echo,noop,user_p]*KeyAB),
  send([A,Na,res,a_user]*KeyAB),
  recv([B,Nb,no_echo,noop,pass_p]*pk(A)),
  send([A,Na,res,a_pass]*pk(B)),
  recv([B,Nb,echo,noop,ui_menu]*pk(A)),
  send([A,Na,res,run]*pk(B)),
  recv([B,Nb,no_echo,pcc]*pk(A)),
  send([A,Na,pcc]*pk(B)),
  recv([B,Nb,no_res,run]*pk(A))
]).

%---------------------------------
%TCB Extension Server is roleB
% shares a symmetric key with the Secure Database Server (SDS)
% which is labeled KeyBD
%---------------------------------
strand(roleB,A,B,D,KeyAB,KeyBD,Na,Nb,Nd,[
  recv([A,Na,sar]*KeyAB),
  send([A,Nb,list]*KeyBD),
  recv([A,Nd,nak]*KeyBD),
  send([B,Nb,echo,noop,user_p]*KeyAB),
  recv([A,Na,res,a_user]*KeyAB),
  send([B,Nb,no_echo,noop,pass_p]*pk(A)),
  recv([A,Na,res,a_pass]*pk(B)),
  send([A,Nb,create,settings]*KeyBD),
  recv([A,Nd,ack]*KeyBD),
  send([B,Nb,echo,noop,ui_menu]*pk(A)),
  recv([A,Na,res,run]*pk(B)),
  send([B,Nb,no_echo,pcc]*pk(A)),
  recv([A,Na,pcc]*pk(B)),
  send([B,Nb,no_res,run]*pk(A))
]).

%---------------------------------
%Secure Database Server (SDS) is roleD
% shares a symmetric key with the TCB Extension Server
% which is labeled KeyBD
%---------------------------------
strand(roleD,A,B,D,KeyAB,KeyBD,Na,Nb,Nd,[
  recv([A,Nb,list]*KeyBD),
  send([A,Nd,nak]*KeyBD),
  recv([A,Nb,create,settings]*KeyBD),
```

```
  send([A,Nd,ack]*KeyBD)
]).


%---------------------------------
%Penetrator
%---------------------------------
strand(test,X,[
  recv(X),
  send(stop)
]).


%---------------------------------
%Demonstration of the trace of the protocols
% There is no penetrator in this run
%---------------------------------

thesisn([Sa,Sb,Sd]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the Nonce from A (the TCBE)
%---------------------------------


thesis0([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,na,St).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the Nonce from B (the TCB Extension
Server)
%---------------------------------


thesis1([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,nb,St).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
```

```
% attempting to check secrecy of the Nonce from D (the Secure Database
Server)
%---------------------------------


thesis2([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,nd,St).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the Symmetric Key shared between A and
B
% (the TCBE and the TCB Extension Server)
%---------------------------------


thesis3([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,keyAB,St).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the Symmetric Key shared between B and
D
% (the TCB Extension Server and the Secure Database Server)
%---------------------------------
thesis4([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,keyBD,St).


%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the user name provided
%---------------------------------
thesis5([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,a_user,St).

%---------------------------------
%Demonstration of the trace of the protocols
% penetrator in this run
% attempting to check secrecy of the password provided
```

```
%---------------------------------
thesis6([Sa,Sb,Sd,St]) :-
  strand(roleA,_A,_B,_D,_KeyAB,keyBD,_Na,nb,nd,Sa),
  strand(roleB,a,b,d,na,keyAB,_KeyBD,_Nb,nd,Sb),
  strand(roleD,a,b,d,na,keyAB,keyBD,nb,_Nd,Sd),
  strand(test,a_pass,St).
```

3. Analysis Output

```
Welcome to SWI-Prolog (Multi-threaded, Version 5.2.13)
Copyright (c) 1990-2003 University of Amsterdam.
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- [csolve_pl,'MLS_LAN_Protocols'].
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:137):
        Singleton variables: [C]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:170):
        Singleton variables: [A, T]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:171):
        Singleton variables: [A, T]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:172):
        Singleton variables: [T]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:302):
        Singleton variables: [T]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:303):
        Singleton variables: [T]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:306):
        Singleton variables: [A]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:312):
        Singleton variables: [Y]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:314):
        Singleton variables: [X]
Warning: (c:/documents and settings/all users/desktop/prolog/workspace/csolve_pl:315):
        Singleton variables: [Y]
% csolve_pl compiled 0.00 sec, 13,288 bytes
Warning: (c:/documents and settings/all
users/desktop/prolog/workspace/mls_lan_protocols:8):
        Singleton variables: [KeyBD, Nd]
Warning: (c:/documents and settings/all
users/desktop/prolog/workspace/mls_lan_protocols:27):
        Singleton variables: [D]
Warning: (c:/documents and settings/all
users/desktop/prolog/workspace/mls_lan_protocols:49):
        Singleton variables: [B, D, KeyAB, Na]
```

% MLS_LAN_Protocols compiled 0.00 sec, 7,404 bytes

Yes
2 ?- thesisn(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 59273 removed for space> --- Try 59274 Try 59275 Try 59276 Try 59277 Try 59278 Try 59279 Try 59280 Try 59281 Try 59282 Try 59283 Try 59284 Try 59285 Try 59286

Simple constraints:
[_G392, [a, b, e]]

Trace:
recv([a, b, _G392])
send([a, _G406, sar]*na)
recv([a, _G406, sar]*na)
send([a, nb, list]*keyAB)
recv([a, nb, list]*keyAB)
send([a, nd, nak]*keyAB)
recv([a, nd, nak]*keyAB)
send([b, nb, echo, noop, user_p]*na)
recv([b, nb, echo, noop, user_p]*na)
send([a, _G406, res, a_user]*na)
recv([a, _G406, res, a_user]*na)
send([b, nb, no_echo, noop, pass_p]*na)
recv([b, nb, no_echo, noop, pass_p]*na)
send([a, _G406, res, a_pass]*na)
recv([a, _G406, res, a_pass]*na)
send([a, nb, create, settings]*keyAB)
recv([a, nb, create, settings]*keyAB)
send([a, nd, ack]*keyAB)
recv([a, nd, ack]*keyAB)
send([b, nb, echo, noop, ui_menu]*na)
recv([b, nb, echo, noop, ui_menu]*na)
send([a, _G406, res, run]*na)
recv([a, _G406, res, run]*na)
send([b, nb, no_echo, pcc]*na)
recv([b, nb, no_echo, pcc]*na)
send([a, _G406, pcc]*na)
recv([a, _G406, pcc]*na)
send([b, nb, no_res, run]*na)
recv([b, nb, no_res, run]*na)

Bundle:
[recv([a, b, _G392]), send([a, _G406, sar]*na), recv([b, nb, echo, noop, user_p]*na), send([a, _G406, res, a_user]*na), recv([b, nb, no_echo, noop, pass_p]*na), send([a, _G406, res, a_pass]*na), recv([b, nb, echo, noop, ui_menu]*na), send([a, _G406, res, run]*na), recv([b, nb, no_echo, pcc]*na), send([a, _G406, pcc]*na), recv([b, nb, no_res, run]*na)]
[recv([a, _G406, sar]*na), send([a, nb, list]*keyAB), recv([a, nd, nak]*keyAB), send([b, nb, echo, noop, user_p]*na), recv([a, _G406, res, a_user]*na), send([b, nb, no_echo, noop, pass_p]*na), recv([a, _G406, res, a_pass]*na), send([a, nb, create, settings]*keyAB), recv([a, nd, ack]*keyAB), send([b, nb, echo, noop, ui_menu]*na), recv([a, _G406, res, run]*na), send([b, nb, no_echo, pcc]*na), recv([a, _G406, pcc]*na), send([b, nb, no_res, run]*na)]
[recv([a, nb, list]*keyAB), send([a, nd, nak]*keyAB), recv([a, nb, create, settings]*keyAB), send([a, nd, ack]*keyAB)]

B = [[recv([a, b, _G392]), send([a, _G406, sar]*na), recv([b, nb, echo|...]*na), send([a, _G406|...]*na), recv([b|...]*na), send([...|...]*na), recv(... *...), send(...)|...], [recv([a, _G406, sar]*na), send([a, nb, list]*keyAB), recv([a, nd|...]*keyAB), send([b|...]*na), recv([...|...]*na), send(... *...), recv(...)|...], [recv([a, nb, list]*keyAB), send([a, nd|...]*keyAB), recv([a|...]*keyAB), send([...|...]*keyAB)]]

Yes
3 ?- thesis0(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
4 ?- thesis1(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
5 ?- thesis2(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
6 ?- thesis3(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
7 ?- thesis4(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
8 ?- thesis5(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try

2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No
9 ?- thesis6(B),search(B,[]).
Starting csolve...
Try 1 Try 2 Try 3 Try 4 Try 5 Try 6 Try 7 Try 8 Try 9 Try 10 Try 11 Try 12 Try 13 Try 14 Try 15 Try 16 Try 17 Try 18 Try 19 Try 20 Try 21 Try 22 Try 23 Try 24 Try 25 Try 26 Try 27 Try 28 Try 29 Try 30 Try 31 Try 32 Try 33 Try 34 Try 35 Try 36 Try 37 Try 38 Try 39 Try 40 Try 41 Try 42 Try 43 Try 44 Try 45 Try 46 Try 47 Try 48 Try 49 Try 50 --- <Try 51 – Try 2882865 removed for space> --- 2882866 Try 2882867 Try 2882868 Try 2882869 Try 2882870 Try 2882871 Try 2882872 Try 2882873 Try 2882874 Try 2882875 Try 2882876 Try 2882877 Try 2882878 Try 2882879 Try 2882880

No

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] Wilson, J.D., A Trusted Connection Framework for Multilevel Secure Local Area Networks, Master's Thesis, Naval Postgraduate School, Monterey, California, June 2000.

[2] Webster's Revised Unabridged Dictionary Version published 1913 by the C. & G. Merriam Co. Springfield, Mass. Under the direction of Noah Porter, D.D., LL.D. This version is copyrighted (C) 1996, 1998 by MICRA, Inc. of Plainfield, NJ, Last edit February 3, 1998. [ftp://ftp.uga.edu/pub/misc/webster/], Accessed April 2004

[3] The American Heritage Dictionary of the English Language, Fourth Edition, Copyright © 2000 by Houghton Mifflin Company. Published by Houghton Mifflin Company. [http://www.eref-trade.hmco.com/], Accessed April 2004

[4] The Free On-line Dictionary of Computing, Editor Denis Howe, [http://www.foldoc.org/], Accessed April 2004.

[5] Needham, R. M. and Schroeder, M. D., "Using Encryption for Authentication in Large Networks", ACM 0001-0782/78/1200-0993, 1978

[6] Meadows, C.A., "Formal Verification of Cryptographic Protocols: A Survey", Advances in Cryptology - Asiacrypt '94, LNSC 917, Springer-Verlag, pp. 133-150, 1995

[7] Shmatikov, V., Symbolic Protocol Analysis, slide 3 [http://www.stanford.edu/class/cs259/lectures/12-Symbolic%20Protocol%20Analysis.pdf], accessed August 2004.

[8] Millen, J.K., "Constraint Solving for Protocol Analysis", The Center for Information Systems Security Studies and Research invited lecture Series, Naval Postgraduate School, [http://cisr.nps.navy.mil/lecturearchive.html], Aug 2004

[9] Meadows, C. A., "Open Issues in Formal Methods for Cryptographic Protocol Analysis," Proceedings of DISCEX 2000, IEEE Computer Society Press, pp. 237-250, January 2000.

[10] Lowe, G., "Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR", Tools and Algorithms for the Construction and Analysis of Systems, vol 1055 of lecture Notes in Computer Science, 147-166. Springer Verlag, 1996.

[11] Denning, D. E. and Sacco, G. M., "Timestamps in Key Distribution Protocols", Communications of the ACM, 24(8):533-536, 1981.

[12] Dolev and Yao A., “On the Security of Public Key Protocols”, IEEE Transactions on Information Theory, 29(2) , 198-208, 1983.

[13] Mao, W., “A Structured Operational Modeling of the Dolev-Yao Threat Model”, Trusted E-Services Laboratory, HP Laboratories Bristol, HPL-2002-218, [http://www.hpl.hp.com/personal/Wenbo_Mao/papers/CCSDY.pdf], August, 2002.

[14] Volpano, D., "Formalization and Proof of Secrecy Properties," Proceedings of the 12th IEEE Computer Security Foundations Workshop, Panel Description, pp. 92-95, June 1999.

[15] Tarigan, A., “Survey in Formal Analysis of Security Properties of Cryptographic Protocol”, [http://antareja.rvs.uni-bielefeld.de/avinanta/Publication/SurveyCrypto/surveycrypto.pdf], May 2002.

[16] Cervesato and Syverson, P. F., “The logic of authentication protocols. In Foundations of Security Analysis and Design”, LNCS 2171, pages 63--136. Springer-Verlag, 2001.

[17] Burrows, M., Abadi, M. and Needham, R., “A Logic of Authentication”, ACM 0734-2071/90/0200-0018, Febuary 1990.

[18] Nessett, D.M., “A critique of the Burrows, Abadi, and Needham Logic”, Operating Systems Review, 24(2):35-38, April 1990

[19] Gong, L., Needham, R. and Yahalom, R., “Reasoning about Belief in Cryptographic Protocols”, In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pages 234-248. IEEE Computer Society Press, 1990.

[20] Brackin, S.H., “A HOL extension of GNY for automatically analyzing cryptographic protocols”, In 9th IEEE Computer Security Foundations Workshop, pages 62-76, IEEE CS Press, June 1996

[21] Syverson, P, and van Oorschot P., “On unifying some cryptographic protocols”, In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pages 14-24. IEEE CS Press, May 1994.

[22] Kailar, R., “Accountability in electronic commerce protocols”, IEEE Transactions on Software Engineering, 5(22):313–328, May 1996.

[23] Wedel, G. and Kessler, V., “Formal semantics for authentication logics”, In E. Bertino, H. Kurth, and G. Martella, editors, Proc. ESORICS'96, pages 219-241. LNCS 1146, 1996

[24] Clarke, E. M. and Wing, J. M., “Formal methods: state of the art and future directions”, ACM Computing Surveys, 28(4):626--643, Dec. 1996

[25] Millen, J.K., “The Interrogator Model”, IEEE 1081-6011/ 1995

[26] Longley, D. and Rigby, S., “An Automatic Search for Security Flaws in Key Management Schemes”, Computers and Security, 11(1):75-90, 1992

[27] Meadows, C.A., “The NRL Protocol Analyzer: An Overview”, Journal of Logic Programming, 26(s):113-131, 1996

[28] Kemmerer, R., “Using Formal Methods to Analyze Encryption Protocols”, IEEE Journal on Selected Areas in Communication, 7(4):448-457, 1989

[29] Meadows, C.A., “Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends”, IEEE Journal on selected areas in communications, 21(1), 2003

[30] Thayer Fabrega, F.J., Herzog, J.C. and Guttman, J., “Strand Spaces: Proving Security Protocols Correct”, Journal of Computer Security, 7(2/3):191-230, 1999

[31] Gritzalis, S., Spinellis, D. and Georgiadis, P., “Security Protocols Over Open Networks and Distributed Systems: Formal methods for their Analysis, Design, and Verification”, Computer Communications, 22(8): 695-707, May 1999

[32] Thayer Fabrega, F.J., Herzog, J.C. and Guttman, J.D., “Strand Space Pictures”, In Proceedings of the Workshop on Formal Methods and Security Protocols, 1998

[33] Thayer Fabrega, F.J., Herzog, J.C. and Guttman, J.D., “Strand Spaces: Why is a Security Protocol Correct?” In Proceedings of the 1998 IEEE Symposium on Security and Privacy, pages 160--171, IEEE Computer Society Press , May 1998

[34] Thayer Fabrega, F.J., Herzog, J.C. and Guttman, J.D., “Mixed Strand Spaces”, In Proceedings of the 12th IEEE Computer Security Foundations Workshop, 27(2): 10-14, IEEE Computer Society Press, June 1999.

[35] J. Zhou, J., “Further Analysis of the Internet Key Exchange Protocol”, Computer Communications 23:1606-1612, Elsevier Science B.V., 2000

[36] Thayer Fabrega, A.L., Herzog, J.C. and Guttman, J.D., “Authentication and Confidentiality via IPSec”, appears in ESORICS, Springer LNCS, MITRE Corporation, June 2000

[37] Mitchell, J. and Shmatikov, V., Presentation: SSL/TLS case study, [www.stanford.edu/class/cs259/lectures/02-SSL.pdf], Jan 8, 2004

[38] Millen, J. K. and Shmatikov, V., “Constraint Solving for Bounded-process Cryptographic Protocol Analysis”, In Procedures of the 8th ACM Conference on Computer and Communications Security (CCS), pages 166--175, 2001.

[39] The SWI-Prolog Website [http://www.swi-prolog.org], Aug 2004.

[40] The Constraint Solving in Prolog Website, [http://www.csl.sri.com/users/millen/capsl/constraints.html], Aug 2004.

[41] Irvine, C., “Assurance Mappings and Formality in Secure Systems”, Class lecture CS4600 L11_FormalWork.pdf, Naval Postgraduate School, Spring 2004.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Dr. George Dinolt
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA

4. Dr. Sylvan S. Pinsky
   NSA
   Fort Meade, MD

5. Dr. Cynthia Irvine
   Computer Science Department
   Naval Postgraduate School
   Monterey, CA

6. Cathy Meadows
   Naval Research Laboratory
   Washington, DC

7. John Millen
   SRI International
   Menlo Park, CA

8. Joshua D. Guttman
   The MITRE Corporation
   Bedford, MA

9. Daniel Craven
   Civilian
   Naval Postgraduate School
   Monterey, CA